SOME CONTRIBUTIONS TO MATHEMATICAL PROGRAMMING

By

PURSHOTAM LAL JOLLY

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AUGUST, 1979

SOME CONTRIBUTIONS TO MATHEMATICAL PROGRAMMING

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

By

PURSHOTAM LAL JOLLY

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AUGUST, 1979

MATH-1979-D-JOL-SOM

CERTIFICATE

Certified that the thesis entitled, "Some Contributions to Mathematical Programming" by Purshotam Lal Jolly has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(S.K.Gupta)
Assistant Professor,
Department of Mathematics,
I.I.T.Kanpur.

August, 1979

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to my thesis supervisor Dr.S.K.Gupta under whose expert guidance this thesis has been written. His constructive criticism and valuable suggestions on the manuscript of the thesis have improved the worth of what little is contained in this thesis. However, I alone should be held responsible for any error or omission thereof.

With great pleasure, I record my gratefulness to Prof.C.R.Bector of The University of Manitoba, Winnipeg,Canada. Apart from the guidance and many timely helps that he had rendered to me during my Ph.D. programme, he had been a constant source of inspiration and a lighthouse for me, even though the distance being a few continents apart. My gratitude far exceeds my powers of expression for the patience and tolerance that he has shown towards me in the face of gross inexperience in the subject on my part.

I acknowledge my indebtness to Prof.J.N.Kapur, Head of the Deptt. of Mathematics and to Dr.B.L.Bhatia for their keen interest and continued encouragements in my work, for many timely helps and helpful advices during my Ph.D. programme. I thank Dr.Punyatma Singh in the similar capacity.

I thank my colleague Mr.H.K.Sharma (going to be Dr.H.K. Sharma soon) for some fruitful conversations over the subject. Thanks are also due to Pujari and Gulati for the same. I also

thank my friends Mr.P.P.Singh,Dr.Sacheti,Dr.Varma,Bhat,Patel and many others for giving me their moral supports on some occasions arising during the course of my programme.

I feel it my pleasant duty to express my most grateful thanks to those who have made my stay in the campus as pleasant as at home. I thank brother Jagir Singh and Bakhshish bhabi for their sacrificing nature and kind affection for me. My warmest thanks are due to the affectionate company of jubilant kids, Harvind, Anju and Indu for many a happy hour. Harvind has also helped sometimes in preparing copies of the first draft. I also express my sincere thanks to Mr. and Mrs. Rachhpal Singh for their great affection for me. The sweet company of charming Baby, Jesu and Pintu is also unforgettable.

My heart-felt and affectionate gratitude go to my brother O.P.Sharma and bhabi Sukhraj for sacrificing some of the interests of their lives for my education and for the constant encouragements that I used to receive from them. I must thank my parents and Rana, Happy and Lucky for the love, affection and long patience of all of them.

Finally I thank Mr.Ashok Bhatia for the patience and efficiency he displayed in typing the thesis and Mr.Upadhaya for careful cyclostyling.

Jolly

August,1979 (P.JOLLY)

To

My Parents

CONTENTS

SYNOPSIS

A Thesis entitled "Some Contributions to Mathematical Programming" submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy by Purshotam Lal Jolly to the Department of Mathematics, Indian Institute of Technology Kanpur.

This thesis discusses pseudo-monotonic programming and its applications, the sensitivity analysis for linear complimentarity problem with applications, bounded variable linear programs and the properties of assignment polytope and traveling salesman polytope.

Chapter 1 is introductory in nature and is intended to bring out the contributions made in this thesis in their proper perspective.

In Chapter 2, a linearization technique to solve programming problems with pseudo-montonic objective functions is given. The algorithm solves the pseudo-montonic program by solving a sequence of finite number of linear programs over convex or non-convex sets (e.g. set of discrete points). The applicability of the algorithm is demonstrated by considering pseudo-monotonic integer programs (all integer and mixed integer case), zero-one programs, transportation, assignment, traveling salesman problems, set covering and set partitioning

problems and programming problems with linear fractional objective functions.

In Chapter 3, we discuss the sensitivity analysis of linear complimentarity problem (LCP)

$$w - Mz = q$$

where $w = (w_1, w_2, \ldots, w_n)$, $z = (z_1, z_2, \ldots, z_n)$, M is $n \times n$ matrix and q is n-vector, using Lemke's complimentary pivot algorithm ["Bimatrix Equilibrium Points and Mathematical Programming", Man.Sci., 11, 1965, pp.681-689.], discussing the changes in vector q, changes in one or more columns of matrix M and simultaneous changes in vector q and matrix M. Sensitivity analysis of convex quadratic programming problem is discussed through its LCP. As an application of the procedure, a simple computational scheme is devised to solve linear fractional functional programs and quadratic fractional functional programs.

In Chapter 4, we propose two methods to solve bounded variable linear programs. The approach used in both these methods is essentially a relaxation approach.

In Chapter 5, we study some aspects of assignment polytope (AP_n) by using an elementary approach. In particular, we study the adjacency of the vertices of the assignment polytope, the structure of some special kinds of its faces, the diameter of AP_n and the edge-connectivity

of the graph of AP_n.

In Chapter 6, we make an attempt to study the adjacency of the vertices of the traveling salesman polytope. A necessary condition for non-adjacency of two vertices of traveling salesman polytope is given. A sufficient condition for non-adjacency is also established. However, the necessary condition is not sufficient and the sufficient condition is not necessary. We also formulate some adjacency and non-adjacency rules to generate adjacent and non-adjacent vertices to a given vertex on traveling salesman polytope and discuss their applications.

CHAPTER - 1

INTRODUCTION

There are many problems in government, defence and industrial organisations which are concerned with picking up of a best alternative from amongest a large number of available alternatives. One of the important class of such problems is the allocation of scarce resources in an optimum manner. Some such problems are to plan the economy of a nation in a most efficient manner, deployment of the armed forces in a way so as to maximize the country's chance of winning a war, to maximize the profit of some organisation or to minimize the waste of some products etc. These problems are so important that no organisation can afford to ignore them. The formulation of such problems and their methods of solutions gave rise to a new class of optimization problems which cannot be solved by the classical methods of optimization based on differential calculus or calculus of variations. Among them one is the class of Mathematical Programming Problems. This thesis makes some contributions to this class of problems.

The main purpose of this chapter is to give a brief survey of theoretical and computational developments in mathematical programming which are relevant to this thesis.

The purpose of this survey is to bring out the contribution more clearly, rather than being exhaustive. To make the survey more precise and understandable the present chapter is divided into five sections. Section 1.1 gives preliminaries and definitions. Section 1.2 gives the classifications of mathematical programs. Section 1.3 gives their brief history of development. Section 1.4 discusses relevant computational techniques used in this thesis and in section 1.5, we discuss contributions of the thesis in brief.

1.1 Preliminaries and Definitions

For the sake of completeness, we give here some of the standard definitions.

Let $f : X \to R$ where $X \subset R^n$ and x^1, x^2 be any two points of X. Let $x^0 \in X$ be such that $x^0 = (1-\alpha)x^1 + \alpha x^2$ for $0 \leq \alpha \leq 1$. Clearly $x^0 \in X$ if X is a convex set. Then we say f is [112]

(i) strictly convex : if X is a convex set and

$$f(x^0) < (1-\alpha)f(x^1) + \alpha f(x^2) \qquad \alpha \neq 0, 1$$

(ii) convex : if X is a convex set and

$$f(x^0) \leq (1-\alpha)\, f(x^1) + \alpha f(x^2)$$

(iii) pseudo-convex : if f is continuously differentiable on an open set containing X and

$$\nabla f(x^1)\,(x^2-x^1) \geq 0 \Rightarrow f(x^2) \geq f(x^1)$$

(iv) <u>explicit quasi-convex</u> : if X is a convex set and

$$f(x^o) < \max \{f(x^1),\ f(x^2)\},\ f(x^1) \neq f(x^2),\ \alpha \neq 0,1$$

(v) <u>quasi-convex</u> : if X is a convex set and

$$f(x^o) \leq \max \{f(x^1), f(x^2)\} \quad ; \quad \text{or}$$

equivalently, f is <u>quasi-convex</u> if it is continuously differentiable on an open set containing X and

$$f(x^2) \leq f(x^1) \Rightarrow \quad \nabla f(x^1)(x^2-x^1) \leq 0$$

Likewise we say that f is strictly concave (cancave, pseudo-cancave, explicit quasi-concave, quasi-concave) if -f is strictly convex (convex, pseudo-convex, explicit quasi-convex, quasi-convex) on X.

(vi) <u>pseudo-monotonic</u> : if f is both pseudo-convex and pseudo-concave.

(vii) <u>quasi-montonic</u> : if f is both quasi-convex and quasi-concave.

Below we give some definitions concerning convex polytopes [41,71].

(i) <u>convex polyhedron</u> : By a convex polyhedron we mean solution set X of the following system of linear constraints

$$Ax = b, \quad x \geq 0$$

where A,b are given mxn and mxl matrices respectively and x is a nxl column vector.

(ii) convex polytope : A bounded convex polyhedron is called a convex polytope.

(iii) extreme point(vertex) : A point $x \in X$ is called an extreme point (vertex) of X iff it cannot be expressed as a convex combination of any other two distinct points of X.

A convex polytope can be regarded as the convex hull of its finite number of extreme points.

(iv) adjacent extreme points : Two extreme points x^1 and $x^2 \in X$ are called adjacent (neighbouring) extreme points on X (or adjacent simply) iff every point on the line segment joining x^1 and x^2 is uniquely expressed as a convex combination of extreme points of X.

(v) edge : The line segment joining any pair of adjacent extreme points of X is called an edge of X.

Thus x^1, x^2 are adjacent on X iff the line segment joining x^1 and x^2 is an edge of X.

We adopt the following notation to indicate the line segment joining x^1 and x^2:

$$[x^1, x^2] = \{x : x = \alpha x^1 + (1-\alpha)x^2,\ 0 \leq \alpha \leq 1\}.$$

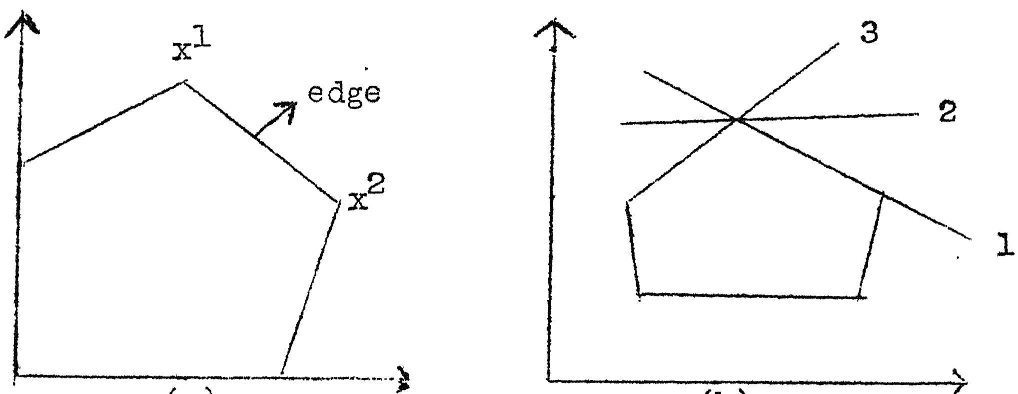
(vi) supporting hyperplane : Consider a boundary point x^1 of convex polytope (or convex set) X. Then $cx = z$ where $c = (c_1, c_2, \ldots, c_n)$ is called a supporting hyperplane at x^1 if $cx^1 = z$ and the whole

of X lies in one closed half space produced by the hyperplane i.e. $cu \geq z \ \forall \ u \in X$ or $cu \leq z \ \forall \ u \in X$.

(vii) <u>face</u> : Face of a polytope is the intersection of the polytope with some of its supporting hyperplane.

(viii) <u>permutation matrix</u> : A nxn zero-one matrix is called a permutation matrix if each of its rows and columns contains exactly one 1.

The above definitions are illustrated in the following figures.



(a) (Convex polytope, extreme pts. and edges)

(b) (Supporting hyperplanes)

(Figure 1)

We now give a few definitions of graph theory [73]. We assume the definitions of a graph, its nodes and of edges.

(i) <u>valency of a node</u> : The valency of a node is the number of edges incident to it.

(ii) connected graph : A graph G is called connected if each node of it can be reached through edges and nodes of it.

(iii) path : A path in a connected graph G is a connected subgraph of the same having two nodes of valency one and all other nodes of valency two.

(iv) length of a path : The length of a path is the number of edges in it.

(v) circuit : A circuit in a graph is a closed path having all nodes of valency two.

1.2 Optimization Problems

The general optimization problem can be described as follows :

Let A be an arbitrary set and B a set with a transitive relation T defined in it. Let $f : A \rightarrow B$. Then the optimization problem is to find an element $x^o \in A$, if it exists, such that

(i) $f(x) \, T \, f(x^o) \quad \forall \quad x \in A - \{x^o\}$

or (ii) $f(x^o) T \, f(x) \quad \forall \quad x \in A - \{x^o\}$

Usually, the set A consists of all possible decisions or policies under consideration, the set B is a subset of real numbers, R, and f is a criterion or objective function i.e. a rule that associates to each decision in A a unique element in B.

With any optimization problem we are primarily concerned with the following types of questions.

(i) Does such an x^o exists ?

(ii) If such an x^o exists, is it unique ?

(iii) If such an x^o exists, is there a procedure to find it out ?

Optimization problems are subdivided into classes of problems either based upon the model or upon the solution procedure.

In the following, we give the various problems which today constitute Mathematical Programming Problems.

1.2.1 Mathematical Programs

A mathematical program is a particular case of the optimization problem. Here set B stands for the set of real numbers, T is the usual ordering ($\leq$, $\geq$) in the set of real numbers and A known as the constraint set, is a subset of R^n which is usually of the form :

$$A = \left\{ x \in R^n \mid g_i(x) \ (\leq \text{ or } \geq \text{ or } =) \ 0, \ i = 1,2,\ldots,m \right\}$$

where $g_i : R^n \rightarrow R$, $i = 1,2,\ldots,m$, are called constraint functions or simply constraints.

Here below we classify the mathematical programs.

1.2.2 Classification of Mathematical Programs

Broadly speaking, the mathematical programs can be classified into two main categories, namely

(A) Stochastic Mathematical Programs : The programs in which the objective function f and/or the constraints g_i's, $i = 1,\ldots,m$, involve elements of chance are called stochastic mathematical programs.

(B) Deterministic Mathematical Programs : The programs which are not stochastic in nature are called deterministic mathematical programs.

We, in this thesis, concentrate on deterministic mathematical programs only.

1.2.3 Classification of Deterministic Mathematical Programs

Depending upon the nature of f and g_i's ($i = 1,..,m$) deterministic mathematical programs can now be broadly classified as under.

B1 Linear Programs : Mathematical programs in which the objective function f and the constraints g_i's($i = 1,\ldots,m$) are linear are called linear programs. Another feature of these programs is the imposition of non-negativity restrictions over the variables x_j ($j = 1,2,\ldots,n$). Linear

programs have large number of applications and are widely studied programs. They can be written as

$$\max \quad f(x) \equiv cx$$

subject to

$$Ax = b$$

$$x \geq 0$$

where c is an n-vector, b an m-vector and A is an mxn matrix.

Linear programs have been studied in more general forms viz.

B11 Semi-Infinite Linear Programs : The linear programs in which m becomes infinite (Charnes, Cooper and Kortanek [32,33,34,35]).

B12 Infinite Linear Programs : The linear programs in which both n and m become infinite (Duffin and Kartovitz [47]).

B2 Non-Linear Programs : A mathematical program which is not linear is called non-linear program. Thus non-linearity can be introduced by taking f and/or one or more of g_i's ($i = 1,\ldots,m$) as non-linear functions. There is a considerable interest in the study of non-linear programs because of their challenging nature. Some of them have been so

extensively studied that they have acquired the status of a separate subject of their own. A few important classes of these programs are listed below :

B21 <u>Convex(Concave)Programs</u> : In these programs we minimize (maximize) convex (concave) - like functions subject to convex/concave - like constraints. As a linear function is convex as well as concave, it can be treated as a convex as well as a concave program. Quadratic programs in which the objective function f is quadratic i.e. $f(x) \equiv x^T Q x + cx$ and g_i's $(i = 1,\ldots,m)$ are linear is another example of such programs. The variables x_j $(j = 1,\ldots,n)$ are assumed to be non-negative. Depending upon the nature of Q, a quadratic program becomes convex or concave.

The other programs in this class include piece-wise convex programs (Zangwill [158], Charnes, Cooper and Kortanek [35]), pseudo-convex programs (Mangasaraian [111]), generalised pseudo-convex programs (Gupta and Bhatt [73]), explicit quasi-convex programs (Bela Martos [115]), quasi-convex programs (Arrow and Enthoven [5]).

B22 Fractional Programs : In these programs, $f \equiv \frac{p}{q}$ where $p : R^n \to R$ and $q : R^n \to R$. If p,q and g_i's $(i = 1,\ldots,m)$ are all linear functions then the program is called linear fractional program otherwise it is the case of non-linear fractional program.

Few important papers concerning fractional programs are (Charnes and Cooper [31] , Dorn [45] , Kanti Swarup [141,142,144,145,146] , Aggarwal [1,2], Dinkelbach [44] , Bector [15,16] , Gogia [65]).

B23 Indefinite Programs : If f is the product of two functions p and q, the mathematical program is called an indefinite program (Bector [16] , Kanti Swarup [145,146] etc.).

B24 Integer Programs : A mathematical program in which the variables are restricted to be integers is called an integer program. If all the variables are restricted to be integers, it is called an all integer program otherwise a mixed integer program (Gomory [66,67]). If the objective function f and constraint set g_i's $(i = 1,\ldots,m)$ are all linear, it is sometimes referred to as a linear integer program. Likewise we may have fractional integer programs and quadratic integer programs.

B25 Zero-One Programs : A mathematical program in which variables are restricted to be zero or one is called zero-one program (Balas [8] , Geoffrion [60] , Glover [61]).

B26 Geometric Programs : A mathematical program in which

$$f(x) = \sum_{j=1}^{N_o} c_{oj} \prod_{i=1}^{n} x_i^{a_{oij}}$$

and the constraint set A is given by

$$A = \left\{ x \in R^n \mid g_k(x) \leq 1,\ k = 1,2,\ldots,m \right\}$$

where $g_k(x) = \sum_{j=1}^{N_k} c_{kj} \prod_{i=1}^{n} x_i^{a_{kij}} \quad \begin{matrix} \leq \\ \equiv \\ \geq \end{matrix} \quad 1$

where N_o represents the number of terms in the objective function, and N_k denotes the number of terms in the kth constraint, c_{oj}'s, c_{kj}'s and a_{kij}'s are real numbers (Duffin [46] , Duffin and Peterson [48] , Passy and Wilde [122]).

1.2.4 Programming Problems in Abstract Spaces :

It is not always that underlying set A is a subset of finite dimensional Euclidean space R^n. A mathematical program in which underlying set A is not a subset of finite dimensional Euclidean space is included in a class of programming problems in abstract spaces. Such programming problems have been studied in locally convex space

(Hurwicz [82] , Hanson [77]), Hilbert space (Kelley and Thompson [94]), pre Hilbert space (Blum [26]).
The following problem in real Banach space has been extensively studied (Varaiya [153]).

$$\max \; f(x)$$

$$\text{subject to } \; g(x) \in A_Y, \; x \in A_X$$

where $A_X \subset X$ and $A_Y \subset Y$, X and Y are real Banach spaces, $g : X \to Y$ and $f : X \to R$.

For further results, refer to (Ritter [131] , Grignard [70] , Neustadt [120], Pchenichney [123]).

1.3 Brief Historical Review

Optimization problems have been of keen interest to mathematicians, engineers and physical scientists since long. Since the end of word war II there has been a rapidly increasing interest in business, management, defence and scientific world in the optimization problems. Although Kantorovitch reported some results in 1939, the real impetus came only in 1947 when G.B.Dantzig gave a very powerful technique called Simplex Method for solving linear programming problems. Although the simplex method was developed in 1947, it was generally not available till 1951, when the cowless commission monograph no. 13 was published.

Soon after that attention was paid to linear programming problems with special structure like transportation, assignment and net work problems etc. and were solved successfully. Charnes and Cooper [30] , Ford and Fulkerson [54,55] and H.W.Kuhn [98] made significant contributions in this direction. The invention of high speed digital computers added more incentive to the developing interest in linear programming problems. David Gale, H.W.Kuhn and A.W.Tucker made important contributions in developing the duality theory for linear programming. C.E.Lemke made a major contribution in this direction by developing a dual simplex technique [103] for solving linear programs.

There are many practical problems which cannot be represented by linear programming models. Hence the need for more general mathematical programming models led to the growing interest in non-linear programming problems. However, the work in this direction started simultaneously. The first set of necessary and sufficient conditions for the existence of optimal solutions to non-linear programming problems came as early as 1951 when H.W.Kuhn and A.W.Tucker [99] published an important paper "Non-linear Programming". Their work laid the foundation of a great deal of later work in non-linear programming.

However, before Kuhn and Tucker, Fritz John [88] derived a set of necessary conditions for the case of inequality constraints alone. The other generalizations include the work by Arrow, Hurvicz and Uzawa [6,7], and Mangasarian [111,113] etc.

The work on integer solutions to linear programming problems also began early in the development of the subject of mathematical programming. Dantzig, Fulkerson and Johnson [43] published a paper in 1954 on the traveling salesman problem which was concerned with the subject of integer solutions to linear programming problem. In 1958, Gomory developed a cutting plane technique, a systematic computational method, for integer solutions to linear programming problems. In 1958, he settled the all integer case [66] and in 1960 a mixed integer case [67]. The methods converge in finite number of iterations and are extensions of dual simplex technique for linear programs. Later on, a number of other methods have appeared. For example, Ben-Israel and Charnes [24], Young [156,157], Fred Glover [62,63,64], Land and Doig [100] etc. Special attention has been paid to the zero-one integer programs. Work of Egon Balas [8], Geoffrion [60] and Glover [61] is significant contribution in this direction. Their methods use implicit enumeration procedure which is essentially a branch and bound procedure.

There are some zero-one programming problems which, because of their special structure, have attracted considerable attention of the research workers in the recent past. Knapsack problem, the traveling salesman problem, set covering and set partitioning problems are few of the important zero-one programs. A few of the important papers in this direction are by Bellman [21] , Bellmore and Nemhauser [22] , Eastman [49] , Little, Murty, Sweeney and Karel [109] ,Held and Karp [79,80], Bellmore and Ratliff [23] , Garfinkel and Nemhauser [57], Lemke, Salkin and Spielberg [107] , Balas and Padberg [9,10]. etc. Cutting plane techniques [23] and implicit enumeration procedures [107,109] are shown to be very effective for the solution of these problems.

The work in quadratic programming also started simultaneously. The most well known methods are by Beale [13,14], Van de Panne and Whinston [150,151,152], Houthakker [81] and Wolfe [155] and thus the quadratic programming problems can be solved by simplex like procedures. The work in linear programming and quadratic programming has attained sufficient maturity by this time. Efforts have been made to provide a unified theory of for studying linear programs and quadratic programs. This link has been established by the complimentary theory of linear programming. Lemke's complimentary pivot method [104]

is a major advance in this direction. Cottle, Dantzig and Eaves have also their major share in developing the theory of complimentarity.

1.4 Relevant Computational Techniques

In Chapters 2,3 and 4 we are concerned with some algorithms for mathematical programs. In this section, we give a brief survey to provide a background of the methods used there.

1. Gomory's Cutting Plane Algorithm

Cutting plane methods have been devised to solve integer linear programs. Using the optimal solution of the linear program (without integer restrictions), if not already an integer solution, a constraint, called a Cut, is obtained which is such that no integer point is cut off and the LP solution is cut off from the original feasible region. The LP is again solved and an additional cut is introduced at each iteration if necessary. Gomory [66,67] has derived cutting planes for all integer solutions and mixed integer solutions to linear programs.

2. Branch and Bound Methods

There are many combinatorial optimization

problems which are very difficult to solve. Some such problems are traveling salesman problems, scheduling problems, Integer programs etc. Branch and bound methods provides a strategy to solve moderate size problems of the above category. The method essentially consists of partitioning the set of feasible solutions into a finite number of subsets and then applying the branch and bound strategy itself to each of the so called candidate problems. The rules which give rise to the subsets are called branching rules. An attempt is also made to obtain a lower (upper) bound for each of the candidate problems. It is obvious that branch and bound methods are problem dependent in the sense that branching rules and bounding rules depend very heavily on the problem itself. The branch and bound strategy has been used very effectively for the following problems :

(i) Traveling Salesman Problem(Little, Murty, Sweeney and Karel [109])

(ii) Scheduling [36]

(iii) Zero-One Programs (Balas [8] , Geoffrion [60] and Glover [61])

(iv) Set Covering Problem (Lemke, Salkin and Spielberg [107])

(v) Set Partitioning Problem (Garfinkel and Nemhauser [57])

3. Lemke's Complimentary Pivot Algorithm

Lemke's pivot algorithm starts with an almost complimentary feasible solution which is obtained by introducing an artificial vector $z_o = (-1,-1,\ldots,-1)$ into the basis. At each stage the algorithm introduces a vector which is complimentary to the vector just dropped from the basis. All the intermediate basic vectors obtained in this algorithm are almost complimentary feasible basic vectors. If at some stage of the algorithm a complementary feasible basis vector is obtained, it is a final basic vector and the algorithm terminates. For details see [119,124].

4. Dinkelbach's Method

Dinkelbach [44] developed a technique to solve the following non-linear fractional program

$$\min_{x \in S} \; N(x)/D(x)$$

where S is compact and connected subset of R^n and $N(x)$ and $D(x)$ are real valued continuous functions on S such that $D(x) > 0$ for all $x \in S$.

The technique depends upon the following theorem :

Theorem 1 : $q_o = N(x_o)/D(x_o) = \max_{x \in S} \; N(x)/D(x)$ iff

$$F(q_o) = F(q_o,x_o) = \max_{x \in S} \; N(x)-q_oD(x) = 0$$

The algorithm is as follows :

Step 1 : Set $q = q_i$ (initially $i = 0$)

Step 2 : Solve $F(q_i) = \min_{x \in S} \left\{ N(x) - q_i D(x) \right\}$

If $F(q_i) = 0$ for $x = x^i$ then x^i solves the fractional program.

Step 3 : If $F(q_i) \neq 0$, compute $q_{i+1} = \dfrac{N(x^i)}{D(x^i)}$ and

go to step 2 with i replaced by i+1.

5. Bounded variable Linear Program (BVLP)

BVLP is of the following type

$$\max \; cx$$

subject to

$$\begin{aligned} Ax &= b \\ x_j &\leq u_j \;, \quad j \in J \\ x_j &\geq 0 \quad \text{for all } j \end{aligned}$$

The simplex method is surely applicable to such a problem but each bounded variable will have to be considered as a separate constraint which results in an increase of the basis size. Taking into consideration the simple nature of the upper bound constraints, Dantzig [40] extended the simplex method itself to solve BVLP without increasing the size of the basis. The details of the method are available in most of the standard text books on linear programming e.g. [76,119,148].

1.5 Contributions of the Thesis

We have give in brief the contributions of the thesis.

1.5.1 Pseudo-monotonic Programs

It is often found in the literature that when an algorithm is devised for linear objective functions, its possible extensions to fractional functions and other non-linear functions is sought. Linear fractional programs where the objective function is a ratio of two linear functions are of great interest. If the numerator represents the total wastage and the denominator represents the cost of production then the objective may be to minimize the wastage per unit production cost. Similarly if the numerator represents the total cost and the denominator represents the volume changes then we may be interested sometimes in minimizing cost per unit volume. Methods for solving linear fractional programs are given by Charnes and Cooper [31], Dinkelbach [44] and Bela Martos [114].

Linear fractional functions are special case of pseudo-monotonic functions methods for which, in general, are not available. In Chapter 2, we develop an algorithm for this more general class of functions. The method is a

linearization technique and provides solution to a pseudo-monotonic program in a finite number of steps by solving a sequence of programs with linear objective functions.

The method is based upon a theorem which is an extension of the following theorem due to Kortanek and Evans [97] .

<u>Theorem 2</u> : Let f be a pseudo-concave function on a closed, convex set X and $x^* \in X$. Consider the two programs :

I max $f(x)$, subject to $x \in X$.

II max $\nabla f(x^*)x$, subject to $x \in X$.

Then x^* is an optimal solution of Program I iff x^* is an optimal solution of Program II.

We extend this theorem for discrete set X which is non-convex.

Our algorithm for pseudo-monotonic programs solves all those pseudo-monotonic programs for which a solution procedure exists in case of linear objective functions. In particular, we have solved pseudo-monotonic integer programs (all integer and mixed integer), pseudo-monotonic zero-one programs, pseudo-monotonic transportation and assignment problems, pseudo-monotonic traveling salesman problem, pseudo-monotonic set covering and set partitioning problems.

1.5.2 Linear Complimentarity Problem

The Linear Complimentarity Problem (LCP) is

$$\text{(LCP)} \qquad \begin{aligned} w - Mz &= q \\ w \geq 0,\ z &\geq 0 \\ w^T z &= 0 \end{aligned}$$

where w, z and q are n-vectors, M is an nxn matrix. There are many problems which can be formulated as LCP. The linear programming problems, convex quadratic programming problems, the problem of computing an equilibrium pair of strategies for bimatrix games are some of the problems which can be formulated as LCP. For example, linear programming problem can be formulated as LCP by combining primal and dual problems and the complimentary slackness property. The primal problem and its dual are

$$\min \ cx$$

subject to

$$\text{(PP)} \qquad \begin{aligned} Ax &\geq b \\ x &\geq 0 \end{aligned}$$

$$\max \ b^T y$$

subject to

$$\text{(DP)} \qquad \begin{aligned} A^T y &\leq c^T \\ y &\geq 0 \end{aligned}$$

If $u \geq 0$, $v \geq 0$ are surplus and slack variables in (PP) and (DP) respectively then at the optimal solution

$$\text{(CS)} \qquad v^T x + u^T y = 0$$

Taking $w = \begin{bmatrix} v \\ u \end{bmatrix}$, $z = \begin{bmatrix} x \\ y \end{bmatrix}$, $q = \begin{bmatrix} c^T \\ -b \end{bmatrix}$ and $M = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}$

the (PP), (DP) and (CS) can be put as

$$w - Mz = q$$

It is to be noted that M is a square, asymmetric, positive semi-definite matrix (PSD) of order (m+n).

Details of the problems which can be formulated as LCP are available in [119,124].

Dantzig and Cottle [37,38] and Lemke [104] have given methods to solve LCP of which Lemke's method is applicable to a large variety of problems. Various conditions have been imposed over the matrix M under which Lemke's method either finds a solution or shows that there is no solution to the problem. If the matrix M is PSD, which is so in case of LCPs corresponding to linear programs and convex quadratic programs, Lemke's method finds a solution if there is any i.e. if the problem is feasible and unbounded, otherwise it will terminate in ray showing that there is no solution.

Details of the algorithm are available in [119,124] . Also various kinds of matrices M have been defined for which it has been shown that LCP has a solution [105] .

In Chapter 3, we discuss the sensitivity analysis of LCP considering changes in elements of q and M using Lemke's method. As an application of the sensitivity procedure, it is shown that linear fractional functional programs and quadratic fractional functional programs can be solved considering sensitivity analysis of corresponding LCPs with changes in q only. A simple computational scheme is devised to solve such problems.

1.5.3 Bounded Variable Linear Programs

Dantzig has proposed a method to solve BVLP restricting the basis size to m only. The methods has the following drawbacks :

(i) Though the criteria for the selection of the vector to enter the basis is the same as in simplex method, the criteria for the vector which leaves the basis is decided by the row in which min $\theta = \{\theta_1, \theta_2, u_j\}$ occurs where $\theta_1 = \min\limits_i \left\{ \frac{x_{B_i}}{y_{ij}} , y_{ij} > 0 \right\}$, $\theta_2 = \min\limits_i \left\{ \frac{u_i - x_{B_i}}{-y_{ij}} , y_{ij} < 0 \right\}$ and u_j is the maximum level (upper bound) to which x_j can be raised. This criteria is pretty involved.

(ii) There are many problems where optimal solution to linear program (without upper bounds) violates only a few upper bound constraints. Dantzig's method does not give us any information about this.

In Chapter 4, we propose two methods to solve BVLP. The proposed methods take care of above drawbacks, to a certain extent. One of our methods makes use of Dantzig's method while the other is completely independent of it. Both the methods start with the optimal solution of linear program (without upper bounds). Our approach is essentially a <u>relaxation approach</u> i.e. we relax some constraints at a time and then introduce them sequentially to arrive at the solution of BVLP.

1.5.4 Assignment Polytope

Assignment polytope which arises from the consideration of the assignment problem is the solution set of the following system of linear equations

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\ldots,n$$

$$x_{ij} \geq 0$$

The extreme points of the assignment polytope of order n (AP_n) are precisely the permutation matrices of order n [92] . It is this consideration which allows the replacement of the condition $x_{ij} = 0$ or 1 in the assignment problem by $x_{ij} \geq 0$ and regards AP_n as convex hull of permutation matrices of order n.

In Chapter 5, we discuss the properties of the assignment polytope. Our approach is completely elementary. In particular, we study the adjacency of the vertices of the assignment polytope, the structure of some special kinds of its faces, the diameter of AP_n and the edge-connectivity of the graph of AP_n. By identifying the vertices of AP_n, which are permutation matrices of order n, with permutations, we prove the following major result.

<u>Theorem 3</u> : Two permutations π and σ are adjacent on AP_n iff $\sigma\pi^{-1}$ is cyclic.

We have further proved that the diameter of the assignment polytope is two and that it is edge-N(n)-connected where

$$N(n) = \sum_{r=2}^{n} {}^{n}c_{r} \, (r-1)!$$

is the number of vertices adjacent to a given vertex on AP_n.

1.5.5 Traveling Salesman Polytope

If in the above system of linear equations we further impose a condition that the solution set of x_{ij}'s (such that $x_{ij} = 0$ or 1) forms a tour where by a tour (or tour solution) we mean a solution of the type for which $x_{i_1i_2} = x_{i_2i_3} = \dots = x_{i_{n-1}i_n} = x_{i_ni_1} = 1$ and zeroes elsewhere, then the related problem is referred to as traveling salesman problem. The associated convex polytope is called the traveling salesman polytope (TP_n) of order n which is the convex hull of cyclic permutation matrices of order n. (By a cyclic permutation matrix we mean the permutation matrix such that the position of one's give a tour solution. Cyclic permutation matrix is identified by a cyclic permutation of order n i.e. the tour on n symbols). TP_n can be regarded as collapsed polytope [118] of order AP_n.

In Chapter 6, we make an attempt to study the adjacency of the vertices of TP_n. The approach used for AP_n in Chapter 5 is extended in the case of TP_n. A necessary condition for two tours to be non-adjacent on TP_n is given. A sufficient condition for two tours to be non-adjacent on TP_n is also given. Counter examples are given to show that neither the necessary condition is sufficient nor the sufficient condition is necessary. Some

adjacency rules and non-adjacency rules are formulated on TP_n which generate tours adjacent and non-adjacent respectively to a given tour. These rules can be used to develop some good heuristic procedures for solving traveling salesman problems. Finally it is shown that TP_n is hamiltonian.

CHAPTER - 2

PSEUDO-MONOTONIC PROGRAMMING AND ITS APPLICATIONS

2.1 INTRODUCTION

It is well known that for a large class of non-linear programmes, there exists a linear programme such that the optimal solutions of the two programmes are same. This is essentially an outcome of the Kuhn Tucker theory. However, such a linear programme is known only in terms of the optimal solution of the non-linear programme. For example, Kortanek and Evans proved the following theorem :

Theorem 1 (Kortanek and Evans [97]): Let f be a pseudo-concave function on a closed, convex set X and $x^* \in X$. Consider the two programs :

Program I maximize $f(x)$, subject to $x \in X$.

Program II maximize $\nabla f(x^*)x$, subject to $x \in X$.

Then x^* is an optimal solution of program I iff x^* is an optimal solution of program II.

Linearization technique for solving such non-linear programmes attempts to solve a sequence of linear programmes in order to arrive at the required linear programme. Bhatt [25] has given a linearization technique for solving pseudo-concave programmes.

In this chapter, we consider pseudo-montonic programmes of the type

$$\text{(P)} \qquad \begin{aligned} &\max \ f(x) \\ &\text{subject to } \ Ax = b \\ &\qquad x \geq 0 \\ &\qquad x \in Q \subset R^n \end{aligned}$$

where A is an mxn matrix of full row rank, b is m-vector and f is pseudo-monotonic function on Q. We also assume that $\{x \mid Ax = b,\ x \geq 0\}$ is bounded. The set Q is such that when f(x) is linear, the solution procedure for solving the problem is available. For example, the set Q may be set of integer solutions or zero - one solutions.

Kortanek and Evans theorem is not valid if programme I is programme P. In view of this, we prove a theorem for program P similar to Kortanek and Evans theorem,in section 2.2. Using this theorem, a linearization algorithm is given in section 2.2 to solve pseudo-monotonic programmes of the above type. In sections 2.3 to 2.7, the linearization technique has been used to solve the following problems with pseudo-monotonic objective functions : all integer and mixed integer programmes, zero - one programmes, transportation, assignment and traveling salesman problems, set covering and set partitioning problems.

Some research workers who have earlier worked on related problems are : Swarup [141,143],Grunspan and Thomas [72],

Florian and Robillard [53], Taha [147], Suresh Chandra [28], Anzai [3], Ishi, Ibaraki and Mine [85], Arora and Puri [4]. Problems solved by them are for fractional objective functions which are special cases of pseudo-monotonic functions [116]. Thus the linearization technique given here is more general and solves a large class of problems.

2.2 The Linearization Algorithm

The proposed algorithm to solve the pseudo-monotonic program (P) is based on the following theorem.

Theorem 2 : Let f be a pseudo-concave function on S where $S = \{x \in R^n \mid Ax = b, x \geq 0\} \cap Q$. Consider the programmes

$$\text{(PI)} \quad \begin{array}{l} \max f(x) \\ \text{subject to } x \in S \end{array}$$

$$\text{(PII)} \quad \begin{array}{l} \max \nabla f(x^*)x \\ \text{subject to } x \in S. \end{array}$$

If x^* is a solution of (PII), then x^* is a solution of (PI).

Proof : x^* is a solution of program II

$$\Rightarrow \nabla f(x^*)x^* \geq \nabla f(x^*)x, \ \forall\ x \in S$$

$$\Rightarrow (x-x^*)\nabla f(x^*) \leq 0, \ \forall\ x \in S$$

$$\Rightarrow f(x) \leq f(x^*), \ \forall\ x \in S \text{ (by pseudo-concavity)}$$

$\Rightarrow$ x^* is a solution of program I.

<u>Theorem</u> 3 : If f is a pseudo-monotonic function on S and x^{i+1} is an optimal solution of

$$(L_i) \quad \begin{array}{l} \max \ \nabla f(x^i)x \\ \text{subject to} \quad x \in S \end{array}$$

such that $x^{i+1} \in \{x^1, x^2, \ldots, x^i\}$, i.e. $x^{i+1} = x^j$ for some $j, 1 \leq j \leq i$, then $x^j, x^{j+1}, \ldots, x^i$ are all optimal solutions of the program (P).

<u>Proof</u> : Without loss of generality, let x^{k+1} solves (L_k), $1 \leq k \leq i$. Then $\nabla f(x^k)x^{k+1} \geq \nabla f(x^k)x$ for all x in S. As $x^k \in S$ therefore $\nabla f(x^k)x^{k+1} \geq \nabla f(x^k)x^k$ which implies $\nabla f(x^k)(x^{k+1} - x^k) \geq 0$.

Since f is pseudo-convex therefore it implies

$$f(x^{k+1}) \geq f(x^k).$$

Hence $f(x^1) \leq f(x^2) \leq \ldots. \leq f(x^i) \leq f(x^{i+1})$. (1)

Now $x^j = x^{i+1}$ maximizes (L_i), for some j, $1 \leq j \leq i$. If $j = i$, that is $x^j = x^i$ then by Theorem 2, x^i solves program (P). If $i \leq j < i$ then

$$\nabla f(x^i)x^j \geq \nabla f(x^i)x \text{ for all } x \in S.$$

Since $x^i \in S$, therefore

$$\nabla f(x^i) \ (x^j - x^i) \geq 0, \quad (2)$$

which implies, $f(x^j) \geq f(x^i)$ (3)

(by pseudo-convexity of f).

From (1) and (3) we get

$$f(x^j) = f(x^{j+1}) = \ldots. = f(x^i) \tag{4}$$

Since pseudo-convexity implies quasi-convexity therefore

$$f(x^j) = f(x^i) \Rightarrow \nabla f(x^i)(x^j - x^i) \leq o, \text{ or}$$

$$\nabla f(x^i)x^j \leq \nabla f(x^i)x^i. \tag{5}$$

Combining (2) and (5) and recalling that $x^j = x^{i+1}$, we get

$$\nabla f(x^i)x^i = \nabla f(x^i)x^{i+1}, \tag{6}$$

which implies that x^i maximizes (L_i).

Now by Theorem 2, x^i is also an optimal solution of (P) and so are the solutions $x^j, x^{j+1}, \ldots., x^{i-1}$ because of (4).

Theorem 4 : If we further assume that the set S is such that $y \in S$ implies $\max_{x \in S} \nabla f(y)x$ exists and the set of optimal solutions is finite then the converse of Theorem 2 is also valid.

Proof: Obvious

Convergence

The algorithm by solving a sequence of linear programs generates a sequence of point of S till one of the points generated earlier is repeated and in view of Theorem 4, finite convergence is granted.

Algorithm :

Using Theorems 2,3 and 4 we now state the algorithm as given below :

Step 1 : Let x^1 be an initial point $\in R^n$ such that $\nabla f(x^1) \neq 0$. Set $i = 1$. Solve the linear program

$$(L_i) \quad \begin{array}{l} \max \ \nabla f(x^i)x \\ \text{subject to} \ \ x \in S \end{array}$$

Let x^{i+1} be a solution of (L_i).

Step 2 : If $x^{i+1} = x^j \in \{x^1, x^2, \ldots, x^i\}$, then $x^j, x^{j+1}, \ldots, x^i$ are solutions of program (P) (Theorem 3). Otherwise, go to step 1 with $i = i+1$.

Remark : If the problem is of minimization in place of maximization the statements and proofs of Theorems 1,2,3 would be similar except that the role of pseudo-convexity and pseudo-concavity gets interchanged and thus the algorithm would work. as such.

In the following sections we show applications of linearization technique to solve some pseudo-monotonic programming problems. The pseudo-monotonic program

$$\begin{array}{c} \max(\min) \ f(x) \\ \text{subject to} \ x \in S \end{array}$$

is referred as pseudo-monotonic integer program, zero-one program, transportation and assignment problem, traveling

salesman problem, set covering and set partitioning problem depending upon the nature of the constraint set. For example the following program is referred as pseudo-monotonic integer program (all integer or mixed integer):

$$\max \ f(x)$$
$$\text{subject to} \ \ x \in S$$

where $S = \left\{ x \in R^n \mid Ax = b,\ x \geq 0,\ x \text{ is integer or mixed integer (as the case may be)} \right\}$

where f is pseudo-monotonic on S and S is bounded.

The fractional programs $\max(cx+\alpha)/(dx+\beta)$ subject to $x \in S$ such that $dx+\beta > 0 \ \forall \ x \in S$ are special cases of pseudo-monotonic programs [116] . At some places we are taking examples with fractional objective functions.

2.3 Pseudo-monotonic Integer Programming

Methods for finding integer solutions to linear fractional functional programs which are special cases of pseudo-monotonic programs are given by Swarup [141], Grunspan and Thomas [72] , Anzaï [3] , Ishii, Ibaraki and Mine [85] . In general, the methods for solving pseudo-monotonic integer programs are not available. In [18,19] an algorithm for pseudo-monotonic integer programs and some other programming problems is given.

In this section we use the methodology developed in section 2.2 to solve the pseudo-monotonic integer programs.

Case 1 : All Integer Pseudo-monotonic Program

Example 1* :

$$\text{(P)} \qquad \max f(x) = \frac{x_1+2-x_2\sqrt{(x_1+2)^2 + x_2^2 - 1}}{(x_1+2)^2 + x_2^2}$$

subject to

$$\begin{aligned} x_1 \qquad + 2x_3 &\leq 3 \\ 2x_2 \qquad &\leq 5 \\ x_3 &\leq 2 \end{aligned}$$

x is all integer $\geq$ 0.

Now for any x

$$(\alpha^2+1)^2\alpha \, \nabla f(x) = \begin{bmatrix} -(x_1+2-x_2\alpha)\left[x_2+(x_1+2)\alpha\right] \\ -\left[x_2+(x_1+2)\alpha\right] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

*The objective function of this example is due to Bela Martos [116].

where $\alpha = \sqrt{(x_1+2)^2 + (x_2)^2} - 1$

and x_4, x_5, x_6 are slack variables. Let S be the solution set.

Iteration 1:

Step 1 : Let $x^1 = (3,0,0,0,5,2)$ be the initial point.

Solve the integer linear program

$$(L_1) \qquad \begin{array}{l} \max \ \nabla f(x^1)x = -25\sqrt{24}\ x_1 - 600x_2 \\ \text{subject to} \quad x \in S \end{array}$$

by Gomory's all integer method [66,74].

$x^2 = (0,0,1,1,5,1)$ is optimal solution of (L_1).

Step 2 : $x^2 \notin \{x^1\}$.

Iteration 2 :

Step 1 : Solve the integer linear program :

$$(L_2) \qquad \begin{array}{l} \max \ \nabla f(x^2)x = -4\sqrt{3}\ x_1 - 12x_2 \\ \text{subject to} \quad x \in S \end{array}$$

$x^3 = (0,0,0,3,5,2)$ is optimal solution of (L_2)

Step 2 : $x^3 \notin \{x^1, x^2\}$.

Iteration 3 :

Step 1 : Solve the integer linear program :

$$(L_3) \qquad \begin{array}{l} \max \ \nabla f(x^3)x = -4\sqrt{3}\ x_1 - 12x_2 \\ \text{subject to } x \in S \end{array}$$

which is same as program (L_2). Its optimal solution is $x^4 = (0,0,0,3,5,2)$.

Step 2 : Since $x^4 = x^3$ and thus the optimal solution of pseudo-monotonic integer program (P) is $x^4 = (0,0,0,3,5,2)$ which gives optimal value of $f(x)$ equal to $\frac{1}{2}$.

Case 2 : Mixed Integer Pseudo-monotonic Program

Example 2 :

$$(P) \quad \max f(x) = \frac{2x_1 - x_2 + 1}{x_1 + x_2 + 2}$$

subject to

$$3x_1 + 7x_2 \leq 20.5$$
$$4x_1 + 5x_2 \leq 21$$
$$x_1, x_2 \text{ integers} \geq 0$$

For any x,

$$\frac{(x_1+x_2+2)^2}{3} \nabla f(x) = \begin{bmatrix} (x_2+1) \\ -(x_1+1) \\ 0 \\ 0 \end{bmatrix}$$

where x_3, x_4 are the slack variables. Take $x^1 = (0,3,0,0)$ as initial pt.

Iteration 1 :

Step 1 : Solve the mixed integer program :

$$(L_1) \quad \begin{array}{l} \max \ \nabla f(x^1)x = 4x_1 - x_2 \\ \text{subject to} \quad x \in S \end{array}$$

by Gomory's method for mixed integer programs [67, 74]

This gives $x^2 = (5,0,5.5,1)$ as optimal solution of (L_1).

Step 2 : $x^2 \notin \{x^1\}$

Iteration 2 :

Step 1 : Solve the mixed integer program :

$$(L_2) \quad \begin{array}{l} \max \quad \nabla f(x^2)x = x_1 - 6x_2 \\ \text{subject to} \quad x \in S \end{array}$$

Solution is $x^3 = (5,0,5.5,1)$

Step 2 : Since $x^3 = x^2$, therefore x^3 is the optimal solution of pseudo-monotonic mixed integer program (P) with optimal objective function value $= \frac{11}{7}$.

2.4 Pseudo-monotonic Zero-One Programs

Balas [8] , Geoffrion [60] and Glover [61] have given methods for solving linear programs with zero-one variables. Florian and Robillard [53] and Taha [147] have given methods for solving zero-one fractional programs. In general, methods for pseudo-monotonic zero-one programs are not known. Linearization technique can be used to solve pseudo-monotonic zero-one programs. This solves, as a special case, the linear fractional zero-one programs also.

Example 3 : Fractional Zero-One Program

$$\min f(x) = \frac{2x_1+3x_2-4x_3-x_4+x_5+1}{x_1+2x_2+x_3+3x_4+2x_5+2}$$

(P)

subject to

$$\begin{array}{l} x_1 + x_2 + x_3 + 2x_4 + x_5 \leq 4 \\ 7x_1 + 3x_3 - 4x_4 + 3x_5 \leq 8 \\ 11x_1 - 6x_2 + 3x_4 - 3x_5 \geq 3 \\ x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,5. \end{array}$$

$\frac{\partial f}{\partial x_j}$ $(j = 1,2,\ldots,5)$, after omission of common positive multiplier $1/(\text{denominator})^2$ are as follows :

$$\frac{\partial f}{\partial x_1} = x_2+6x_3+7x_4+3x_5+3$$

$$\frac{\partial f}{\partial x_2} = -x_1+11x_3+11x_4+4x_5+4$$

$$\frac{\partial f}{\partial x_3} = -6x_1-11x_2-11x_4-9x_5-9$$

$$\frac{\partial f}{\partial x_4} = -7x_1-11x_2+11x_3-5x_5-5$$

$$\frac{\partial f}{\partial x_5} = -3x_1-4x_2+9x_3+5x_4$$

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_5}\right)$$

Iteration 1 :

Step 1 : $x^1 = (1,1,0,0,0)$ is initial point.

Solve the linear zero-one program :

$$(L_1) \qquad \begin{array}{l} \min \ \nabla f(x^1)x = x_1 - x_2 - 17x_3 - 18x_4 - 7x_5 \\ \text{subject to} \quad x \in S \end{array}$$

by Balas' additive algorithm for zero-one programs [8,60,148].

This gives $x^2 = (0,0,1,1,0)$ as the optimal solution.

Step 2 : $x^2 \notin \{x^1\}$

Iteration 2 :

Step 1 : Solve the linear zero-one program :

$$(L_2) \qquad \begin{array}{l} \min \ \nabla f(x^2)x = 13x_1 + 22x_2 - 11x_3 + 11x_4 + 14x_5 \\ \text{subject to} \quad x \in S. \end{array}$$

We get $x^3 = (0,0,1,1,0)$

Step 2 : Since $x^3 = x^2$, therefore x^3 is the optimal solution of pseudo-monotonic program (P) with optimal objective function value $= -\frac{2}{3}$.

Remark 1: The initial point $x^1 = (1,1,0,0,0)$ was the worst solution with objective function value $= \frac{6}{5}$.

Example 4 :

$$\min f(x) = \frac{x_1 + 2 - x_2\sqrt{(x_1+2)^2 + x_2^2 - 1}}{(x_1+2)^2 + x_2^2}$$

subject to

$$x_1 + x_2 \leq 2$$

$$x_2 \leq 1$$

$$x_1, x_2 = 0 \text{ or } 1$$

is a pseudo-monotonic zero-one program. It can be seen that this problem has two optimal solutions, namely (1,1) and (0,1) with minimum value of objective function being equal to zero.

2.5 Pseudo-monotonic Transportation and Assignment Problems

Swarup [143] and Suresh Chandra [28] have given methods for solving transportation problems and assignment problems with linear fractional objective function. Linearization technique can be used to solve transportation and assignment problems with pseudo-monotonic objective functions which, as a special case, solves fractional transportation and assignment problems also. For the purpose of illustration we present fractional programs.

Example 5 : Fractional Transportation Problem

$$\text{(P)} \qquad \min \ f(x) = \frac{\sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} x_{ij}}{\sum_{i=1}^{3} \sum_{j=1}^{4} d_{ij} x_{ij}}$$

subject to usual linear transportation constraints with availabilities 10,8,6 and demands 7,8,5,4. The cost matrices $\{c_{ij}\}$ and $\{d_{ij}\}$ are given below :

$$\{c_{ij}\} = \begin{array}{|c|c|c|c|} \hline 2 & 3 & 4 & -1 \\ \hline 1 & 2 & -1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline \end{array}$$

$$\{d_{ij}\} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 2 & 1 \\ \hline 1 & 3 & 1 & 2 \\ \hline 2 & 1 & 1 & 2 \\ \hline \end{array}$$

We take $x^1 \equiv x^1_{ij}$, the initial basic feasible solution given by north-west corner rule [76], as the initial point for the algorithm and is given below :

(Tableau 1)

24	7	8	5	4
10	7	3		
8		5	3	
6			2	4

The circled entries denote the values of basic cells, the first row and first column denote demands and availabilities.

Iteration 1 :

Step 1 : Solve the linear transportation problem :

$$(L_1) \qquad \begin{array}{c} \min \ \nabla f(x^1)x \\ \text{subject to } x \in S \end{array}$$

where S is the corresponding transportation polytope. The cost matrix $\{\nabla f(x^1)\}$ for the problem is given in tableau 2.

(Tableau 2)

$$\{\nabla f(x^1)\} = \begin{array}{|c|c|c|c|} \hline 48 & 86 & 96 & -66 \\ \hline 10 & -8 & -66 & -18 \\ \hline -18 & -66 & 10 & -94 \\ \hline \end{array}$$

Dantzig's uv method [76] gives the optimal transportation solution x^2 as given in tableau 3.

(Tableau 3)

24	7	8	5	4
10	6			4
8	1	2	5	
6		6		

Step 2 : $x^2 \notin \{x^1\}$

Iteration 2 :

Step 1 : Solve the linear transportation problem :

$$(L_2) \quad \begin{array}{l} \min \ \nabla f(x^2)x \\ \text{subject to } \ x \in S \end{array}$$

The cost matrix $\{\nabla f(x^2)\}$ is given in tableau 4.

(Tableau 4)

$\{\nabla f(x^2)\}$ =

54	82	108	-30
26	50	-30	24
24	-30	26	-32

The optimal solution x^3 of (L_2) is given in tableau 5.

(Tableau 5)

24	7	8	5	4
10	6			4
8	1	2	5	
6		6		

Step 2 : Since $x^3 = x^2$, therefore x^3 is the optimal solution of pseudo-monotonic transportation problem (P) with optimal value $= \frac{1}{14}$.

Example 6 : The Fractional Assignment Problem

$$(P) \qquad \min \quad f(x) = \frac{\sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij} x_{ij}}{\sum_{i=1}^{4} \sum_{j=1}^{4} d_{ij} x_{ij}}$$

subject to usual linear assignment constraints,

where the cost matrices $\{c_{ij}\}$ and $\{d_{ij}\}$ are given below:

$\{c_{ij}\}$ =

3	1	1	-2
1	3	2	-1
2	-3	3	2
1	2	-1	2

$\{d_{ij}\}$ =

1	1	1	1
2	1	1	2
1	2	1	1
2	1	1	1

The extreme points of the assignment polytope of order n are the permutation matrices of order n. Corresponding to a permutation $\pi \in S_n$, the symmetric group of order n, we can associate a permutation matrix $X(\pi) = \{x_{ij}(\pi)\}$ as follows :

$$x_{ij}(\pi) = 1 \quad \text{iff } \pi(i) = j$$
$$= 0 \quad \text{otherwise.}$$

Furthermore, we mean by the symbol $i_1 i_2 \ldots i_n$ as the permutation $\begin{pmatrix} 1 & 2 \ldots n \\ i_1 & i_2 \ldots i_n \end{pmatrix}$. Henceforth we represent the assignment solution by a permutation in one row only (base row understood in its natural order).

Iteration 1 :

Step 1 : Let $\pi^1 = 1234$ be the initial assignment.

Solve the linear assignment problem :

$$(L_1) \quad \begin{array}{l} \min \ \nabla f(X(\pi^1)) \, x_{ij} \\ \text{subject to} \quad x \in S \end{array}$$

where S is the assignment polytope of order 4 whose cost matrix is as given in tableau 6.

(Tableau 6)

1	-7	-7	-19
-18	1	-3	-26
-3	-34	1	-3
-18	-3	-15	-3

The Hungarian assignment technique [98] gives $\pi^2 = 4123$ as the optimal assignment to (L_1).

Step 2 : $\pi^2 \notin \{\pi^1\}$.

Iteration 2 :

Step 1 : Solve the linear assignment problem :

$$(L_2) \qquad \begin{array}{c} \min \ \nabla f(X(\pi^2)) \, x_{ij} \\ \text{subject to } x \in S \end{array}$$

whose cost matrix is given in tableau 7.

(Tableau 7)

23	11	11	-7
16	23	17	4
17	-8	23	17
16	17	-1	17

Optimal assignment to (L_2) is $\pi^3 = 4123$

Step 2 : Since $\pi^3 = \pi^2$, therefore optimal assignment solution of pseudo-monotonic assignment problem (P) is $\pi^3 = 4123$ with optimal objective function value $= -\frac{5}{6}$.

Remark 2 : The initial assignment $\pi^1 = 1234$ with which we started was the worst assignment with value $2\frac{3}{4}$.

2.6 Pseudo-monotonic Traveling Salesman Problem

The traveling salesman problem of order n with linear objective function is written as

$$\text{minimize} \quad f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$(P_o) \qquad \text{subject to} \quad \sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1,2,\ldots,n)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1,2,\ldots,n)$$

$$x_{ij} = 0 \text{ or } 1$$

such that the solution set of those x_{ij}'s for which $x_{ij} = 1$ forms a tour where a tour is a sequence of integers taken from $(1,2,\ldots,n)$ in which each of the integers appears once and only once and we mean by a tour $\pi = (i_1, i_2, \ldots, i_p, \ldots i_n)$ the solution $x_{i_1 i_2} = x_{i_2 i_3} = \ldots = x_{i_p i_{p+1}} = \ldots = x_{i_n i_1} = 1$ and vice versa. Therefore a tour on n symbols can be represented by a cyclic permutation on n symbols and thus

there are $(n-1)!$ tours which are all feasible solutions of the problem (P_o).

The usual terminology is that the n integers correspond to n cities or nodes of a graph, $\{c_{ij}\}$ is the "distance matrix" giving "distances" from node i to j or the arc lengths i to j, a tour π being a closed path passing through each node exactly once and the objective function gives the sum of the arc lengths over the arcs included in the tour.

Traveling salesman problem arises in number of contexts. To name a few of them are routing problems, job scheduling, computer wiring[22,108] and minimizing wall paper wastage [56] .

The proposed traveling salesman problem with pseudo-monotonic objective function is

$$(P) \qquad \begin{array}{l} \min \quad f(x) \\ \text{subject to } x \in TP_n \end{array}$$

where TP_n is traveling salesman polytope of order n which is the convex hull of the cyclic permutation matrices and is the convex feasible set of the problem without integer restrictions.

There are several methods, exact as well as heuristics, available for solving the traveling salesman problem with linear objective function. We essentially need the exact methods for our purpose. See [22] for brief

discussion of them. In general, the methods for pseudo-monotonic traveling salesman problems are not available . We here use the linearization technique to solve such problems. The branch and bound method of little et al [109] was used to solve linearized problems.

Example 7 : Linear Fractional Traveling Salesman Problem

$$\min \quad f(x) \quad = \frac{\sum_{i=1}^{6} \sum_{j=1}^{6} c_{ij}x_{ij} + \alpha}{\sum_{i=1}^{6} \sum_{j=1}^{6} d_{ij}x_{ij} + \beta}$$

subject to usual traveling salesman constraints such that $\sum_{i=1}^{6} \sum_{j=1}^{6} d_{ij}x_{ij} + \beta > 0$ for all tours $\in P_6$, is an example of 6x6 pseudo-monotonic traveling salesman problem. α and β in the present case are taken to be zero matrices. The matrices $\{c_{ij}\}$ and $\{d_{ij}\}$ are given below :

$\{c_{ij}\}$ =

1	2	1	-1	3	2
-1	-2	-3	1	2	3
4	-2	1	-1	2	-2
1	2	3	4	-1	-2
2	-1	-2	-3	1	2
2	3	-1	-4	-5	-6

$$\{d_{ij}\} = \begin{array}{|c|c|c|c|c|c|} \hline 2 & 1 & 2 & 1 & 4 & 5 \\ \hline 1 & 3 & 1 & 4 & 1 & 3 \\ \hline 4 & 3 & 2 & 4 & 5 & 6 \\ \hline 3 & 2 & 5 & 1 & 6 & 2 \\ \hline 1 & -1 & -1 & -1 & -1 & -1 \\ \hline 5 & 2 & 3 & 1 & -1 & -1 \\ \hline \end{array}$$

<u>Iteration</u> 1 :

<u>Step 1</u> : Let $x^1 = (1,2,3,4,5,6)$ be the initial tour.

Solve the linear traveling salesman problem :

$$(L_1) \qquad \begin{array}{l} \min \ \nabla f(x^1)x \\ \text{subject to } x \in P_6 \end{array}$$

for which the distance matrix $\{\nabla f(x^1)\}$ is given in tableau 8, by branch and bound method of Little et al [109].

The optimal tour to (L_1) is $x^2 = (1,4,6,5,3,2)$.

<u>Step 2</u> : $x^2 \notin \{x^1\}$

<u>Iteration</u> 2 :

<u>Step 1</u> : Solve the traveling salesman problem :

$$(L_2) \qquad \begin{array}{l} \min \ \nabla f(x^2)x \\ \text{subject to } x \in P_6 \end{array}$$

for which the distance matrix $\{\nabla f(x^2)\}$ is given in tableau 9.

(Tableau 8)

$\left\{\nabla f(x^1)\right\}$ =

14	31	14	-17	44	27
-17	-35	-49	12	31	45
60	-35	14	-20	27	-38
13	30	43	63	-22	-34
31	-15	-31	-47	17	33
27	46	-19	-65	-79	-95

(Tableau 9)

$\left\{\nabla f(x^2)\right\}$ =

31	23	31	8	67	75
8	29	-2	57	23	54
72	29	31	47	75	68
44	36	80	26	73	16
23	-18	-23	-28	-8	-3
75	41	34	-7	-38	-43

The optimal tour to (L_2) is $x^3 = (1,4,6,5,3,2)$.

Step 2 : Since $x^3 = x^2$, therefore x^3 is the optimal tour to pseudo-monotonic traveling salesman problem (P) which gives optimal objective function value equal to $-\frac{13}{5}$.

Remark 3: The above example was also tried by subtour elimination method of Eastman [49] by starting with same initial solution $x^1 = (1,2,3,4,5,6)$. But, however, the first assignment solutions in both iterations applied to tableau 8 and 9 gave the tours $x^2 = (1,4,6,5,3,2)$ and $x^3 = x^2$.

2.7 Pseudo-monotonic Set Covering and Set Partitioning Problems

The set covering problem (SC) with linear objective function is :

$$\min z = cx$$

subject to

$$Ex \geq e$$

$$\text{and} \quad x_j = 0 \text{ or } 1 \ (j = 1,\ldots,n)$$

where $E = \{e_{ij}\}$ is a mxn matrix of zero's and one's, e is m-column vector of all 1's. The usual interpretation is that the columns of E can be thought of as sets and the rows of E as elements such that

$$e_{ij} = 1 \quad \text{if ith element} \in \text{set } j$$
$$= 0 \quad \text{otherwise}$$

The SC problem is to find a cheepest union of sets from E

that covers every element of e.

If the inequality constraints are replaced by equality constraints the corresponding problem is set partitioning problem (SP). Thus in SP problem, we seek a cheapest union of disjoint sets from E which covers e. SC and SP problems have many practical applications. For details refer to [58,135].

Balas and Padberg [9], Bellmore and Ratliff [23], Etcheberry [52], Garfinkel and Nemhauser [57], Jenson [86], Lawler [101], Lemke, Salkin and Spielberg [107], Salkin and Koncal [136,137] have given methods for SC and SP with linear objective functions.

There are some models where the SC and SP problems with fractional functions arise. For example the locational models where the objective may be to minimize the ratio of cost incurred to some measure of public facility. For example the construction of rationing depots in a big city where a particular depot is assigned to demands of users of particular sectors only. The public utility of the depot depends upon the number of persons who find convenient to visit it. For details of locational models refer to [130,149].

Recently, Arora and Puri [4] have given an enumerative technique for SC for those fractional objective functions in which the cost coefficients c_j's of the numerator are non-negative. In general, the methods for pseudo-monotonic SC and SP have not been given. The linearization technique is used here to

solve SC and SP with pseudo-monotonic objective functions. This solves, as a special case, the SC and SP with linear fractional functions also. Furthermore, unlike Arora and Puri [4], there is no restriction on the cost coefficients c_j's. We give below the solutions to two examples, one each for SC and SP.

Example 8 : Fractional Set Covering Problem

$$(P) \quad \min f(x) = \frac{2x_1+x_2+3x_3+2x_4+x_5+x_6+2x_7+4x_8+2x_9+2x_{10}}{2x_1+2x_2+2x_3+3x_4+3x_5+3x_6+4x_7+4x_8+4x_9+3x_{10}+1}$$

subject to $x \in S$

where $S = \left\{ x \in R^n \mid Ex \geq e,\ x \text{ is zero-one vector} \right\}$ and matrix E is given below :

$$E = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\frac{\partial f}{\partial x_j}$ (j = 1,2,...,10), after omission of common positive multiplier $\frac{1}{(\text{Dominator})^2}$ are as follows :

$$\frac{\partial f}{\partial x_1} = 2x_2-2x_3+2x_4+4x_5+4x_6+4x_7+4x_9+2x_{10}+2$$

$$\frac{\partial f}{\partial x_2} = -2x_1-4x_3-x_4+x_5+x_6-4x_8-x_{10}+1$$

$$\frac{\partial f}{\partial x_3} = 2x_1+4x_2+5x_4+7x_5+7x_6+8x_7+4x_8+8x_9+5x_{10}+3$$

$$\frac{\partial f}{\partial x_4} = -2x_1+x_2-5x_3+3x_5+3x_6+2x_7-4x_8+2x_9+2$$

$$\frac{\partial f}{\partial x_5} = -4x_1-x_2-7x_3-3x_4-2x_7-8x_8-2x_9-3x_{10}+1$$

$$\frac{\partial f}{\partial x_6} = \frac{\partial f}{\partial x_5}, \quad \frac{\partial f}{\partial x_7} = 2\frac{\partial f}{\partial x_2}, \quad \frac{\partial f}{\partial x_8} = 2\frac{\partial f}{\partial x_1}, \quad \frac{\partial f}{\partial x_9} = 2\frac{\partial f}{\partial x_2}, \quad \frac{\partial f}{\partial x_{10}} = \frac{\partial f}{\partial x_4}$$

<u>Iteration 1</u> :

<u>Step 1</u> : Let $x^1 = (1,1,1,1,1,1,1,1,1,1)$ be initial point of (P). Solve the linear SC problem :

$$(L_1) \quad \begin{array}{l} \min\ \nabla f(x^1)x = 22x_1-9x_2+53x_3+2x_4-29x_5-29x_6-18x_7+44x_8-18x_9+2x_{10} \\ \text{subject to } x \in S \end{array}$$

where S is the feasible region of SC, using Lemke, Salkin and Spielberg's [107,135] enumeration procedure.

The solution is $x^2 = (0,1,0,1,1,1,1,0,1,1)$.

<u>Step 2</u> : $x^2 \notin \{x^1\}$

Iteration 2 :

Step 1 : Solve $\nabla f(x^2)x = 24x_1+x_2+47x_3+13x_4-10x_5-10x_6+2x_7$
$+48x_8+2x_9+13x_{10}$

(L_2)

subject to $x \in S$.

The solution is $x^3 = (0,0,0,1,1,1,1,0,0,1)$

Step 2 : $x^3 \notin \{x^1, x^2\}$

Iteration 3:

Step 1 : Solve min $\nabla f(x^3)x = 18x_1+x_2+35x_3+10x_4-7x_5-7x_6+2x_7$
$+36x_8+2x_9+10x_{10}$

(L_3) subject to $x \in S$.

The solution is $x^4 = (0,0,0,1,1,1,1,0,0,1)$

Step 2 : Since $x^4 = x^3$, therefore x^4 is the optimal solution of pseudo-monotonic SC (P) with optimal objective function value equal to $\frac{8}{17}$.

Remark 4 : In SC problem it is assumed without loss of generality that the costs are positive, because non-positive costs can be handled by setting $x_j = 1$ for $c_j \leq 0$ and deleting row i such that $a_{ij} = 1$. Thus, because of the appearance of the negative costs in the objective function of the linearized problems, practically it is seen that size of the reduced problems with positive costs is considerably less and many of them get solved just by inspection.

Example 9 : Fractional Set Partitioning Problem

$$\min f(x) = \frac{x_1+2x_2+x_3+3x_4+x_5+x_6+2x_7+x_8+2x_9+4x_{10}}{2x_1+x_2+2x_3+x_4+3x_5+4x_6+x_7+2x_8+3x_9+x_{10}+1}$$

(P)

subject to $x \in S$.

where $S = \left\{ x \in R^n \,|\, Ex = e, \text{ x is zero-one vector} \right\}$ and the matrix E is :

$$E = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$\frac{\partial f}{\partial x_j}$ $(j = 1,\dots,10)$, after omission of common positive multiplier $1/(\text{Denominator})^2$ are as follows :

$$\frac{\partial f}{\partial x_1} = -3x_2-5x_4+x_5+2x_6-3x_7-x_9-7x_{10}+1$$

$$\frac{\partial f}{\partial x_2} = 3x_1+3x_3-x_4+5x_5+7x_6+3x_8+4x_9-2x_{10}+2$$

$$\frac{\partial f}{\partial x_3} = \frac{\partial f}{\partial x_1}$$

$$\frac{\partial f}{\partial x_4} = 5x_1+x_2+5x_3+8x_5+11x_6+x_7+5x_8+7x_9-x_{10}+3$$

$$\frac{\partial f}{\partial x_5} = -x_1-5x_2-x_3-8x_4+x_6-5x_7-x_8-3x_9-11x_{10}+1$$

$$\frac{\partial f}{\partial x_6} = -2x_1-7x_2-2x_3-11x_4-x_5-7x_7-2x_8-5x_9-15x_{10}+1$$

$$\frac{\partial f}{\partial x_7} = \frac{\partial f}{\partial x_2}, \quad \frac{\partial f}{\partial x_8} = \frac{\partial f}{\partial x_1}$$

$$\frac{\partial f}{\partial x_9} = x_1-4x_2+x_3-7x_4+3x_5+5x_6-4x_7+x_8-10x_{10}+2$$

$$\frac{\partial f}{\partial x_{10}} = 7x_1+2x_2+7x_3+x_4+11x_5+15x_6+2x_7+7x_8+10x_9+4$$

<u>Iteration</u> 1 :

<u>Step 1</u> : Let $x^1 = (0,0,0,0,1,0,1,1,0,0)$ be initial point of (P). Solve the linear SP

$$(L_1) \quad \begin{aligned} &\min \nabla f(x^1)x = -2x_1+8x_2-2x_3+14x_4-6x_5-10x_6+8x_7-2x_8+20x_{10} \\ &\text{subject to } x \in S. \end{aligned}$$

by enumeration procedure of Garfinkel and Nemhauser [57] .

The solution is $x^2 = (0,0,1,0,0,1,0,0,1,0)$

<u>Step 2</u> : $x^2 \notin \{x^1\}$

<u>Iteration</u> 2 :

<u>Step 1</u> : Solve

$$(L_2) \quad \begin{aligned} &\min \nabla f(x^2)x = 2x_1+16x_2+2x_3+26x_4-2x_5-6x_6+16x_7+2x_8+8x_9+36x_{10} \\ &\text{subject to } x \in S. \end{aligned}$$

The solution is $x^3 = (0,0,1,0,0,1,0,0,1,0)$.

<u>Step 2</u> : Since $x^3 = x^2$, therefore x^3 is the optimal solution of pseudo-monotonic SP problem (P) with optimal objective function value equal to $\frac{2}{5}$.

CHAPTER - 3

SENSITIVITY ANALYSIS OF LINEAR COMPLEMENTARITY PROBLEM WITH APPLICATIONS*

3.1 Introduction

The linear Complementarity Problem (LCP) is to find a solution of the system of equations

$$\begin{aligned} & w = Mz + q \\ & w \geq 0,\ z \geq 0 \qquad (1) \\ & w^T z = 0 \end{aligned}$$

where $w = (w_1, w_2, \ldots, w_n)$; $z = (z_1, z_2, \ldots, z_n)$, M is a square matrix order n and q is an n-vector. LCP is closely connected with mathematical programming, game theory, economic equilibrium theory and fixed point theory.

LCP was first studied by Dantzig and Cottle [42] in 1967 in the field of mathematical programming. They proved that necessary optimality conditions for a quadratic programming problems given by Kuhn-Tucker [99] become the LCP and for convex programming problems the Kuhn-Tucker conditions are sufficient as well. Also see [37,38]. They also provided an algorithm for solving LCP which was referred to as the principal pivoting method.

*Presented in Joint National Meeting of ORSA/TIMS held at Los Angles (U.S.A) in Nov.1978 [20].

Another algorithm for LCP is Lemke's algorithm which was originally developed by Lemke and Howson [106] for solving LCP corresponding to the problem of computing an equilibrium pair of strategies in bimatrix game problems which was later extended by Lemke [104] to solve general linear complementarity problems. See [119,124] for details of Lemke's Method. The third method is due to Graves [69].

All the above three methods [37,69,104] are pivoting algorithms and are also referred to as complementary pivoting algorithms. Under certain conditions on the matrix M, these algorithms solve the problem efficiently. Of the three algorithms, Lemke's algorithm is applicable to the broadest class of problems. Cottle and Dantzig [37] in 1968, Lemke [105] in 1970, Eaves [50] in 1971 and Saigal [134] in 1972 etc. have considered some conditions on matrices M under which Lemke's algorithm either leads to a solution or shows that the problem has no solution. Several types of matrices M have been defined for which it can be shown that LCP has a solution. For details refer to [105]. Lemke's algorithm has several applications in linear programming (Ravindran [127]), quadratic programming (Eaves [51]), non co-operative games (Wilson [154], Rosenmüller [133]). Since Lemke's algorithm, a lot of combinatorial methods have

been developed by using his idea. For example see Scarf's [138] method and its applications [139,140].

The second class of algorithms for the LCP is the class of cutting-plane type or branch and bound type algorithms. Kirchgässner [95] has developed a cutting plane algorithm while Jeroslow [87] has characterized all the valid cutting planes for this class of problems. Ibaraki [83,84] has applied branch and bound concepts to solve LCP. These algorithms can solve LCP without any conditions, but their computational efficiency may be questionable. So far more interest has been centred on pivoting algorithms and their degree of efficiency is found comparable to that of the simplex method for solving an LP of a comparable size. The computational experience of Ravindran [128] reveals the superiority of Lemke's complementary pivot method (Lemke's method in short) over the simplex method. Recent studies by Polito [125] and Lee [102] have shown the superiority of Lemke's method for solving convex quadratic programming problems as well. Thus LCP can serve as a unified theory for studying linear and quadratic programs and bimatrix games. Besides these applications, LCP is used to study minimum distance problem, problems in the plastic analysis of structures [110], in the elastic flexural behaviour of reinforced concrete beams, in the free boundary problems for journal bearings, in the

study of finance models and in several other areas. See [37,119,124] for these applications.

The post optimality analysis and sensitivity analysis are fully developed aspects of linear programming. Need for such a analysis often arises in operations research problems because the model parameters used thereof are many times rough estimates subject to some ranges of variations or sometimes as a result of some policy decisions they need to be changed and consequences of such changes are to be viewed. In this chapter, we discuss the sensitivity analysis of LCP through Lemke's complementary pivot method. The following changes have been discussed.

(i) Changes in right hand vector q.

(ii) Changes in one or more column vectors of M.

(iii) Simultaneous changes in q and in one or more column vectors of M.

It is observed that changes in cost coefficients and requirement vector of linear programs and convex quadratic programs can be more conveniently discussed through LCP.

As an application of this procedure, we have further shown that linear fractional functional programs and quadratic fractional functional programs can alternatively be solved by considering the sensitivity analysis of their corresponding LCP's by considering changes in q only.

In section 3.2 we have discussed the sensitivity analysis of LCP. Section 3.3 illustrates the procedure by an example considering the sensitivity analysis of convex quadratic programs. Section 3.4 gives the applications of this analysis to solve linear fractional functional programs and quadratic fractional functional programs.

3.2 Sensitivity Analysis Of Linear Complementarity Problem

Sensitivity analysis of LCP is considered by considering the changes of the following types.

(i) Changes in q vector.

(ii) Changes in one or more column vectors of M.

(iii) Simultaneous changes in q vector and in one or more column vectors of M.

We discuss each of these changes one by one.

(i) Changes in q vector

The initial tableau form for the LCP(1) is given as below :

(Tableau 1)

Basis	$w_1 \ldots w_s \ldots w_n \; z_1 \ldots z_s \ldots z_n$	z_0	q	Δq	$q+\Delta q$
w_1	$1 \ldots 0 \ldots 0 \; -m_{11} \ldots -m_{1s} \ldots -m_{1n}$	-1	q_1	Δq_1	$q_1+\Delta q_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_s	$0 \quad 1 \quad 0 \; -m_{s1} \quad -m_{ss} \quad -m_{sn}$	-1	q_s	Δq_s	$q_s+\Delta q_s$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_n	$0 \quad 0 \quad 1 \; -m_{n1} \quad -m_{ns} \quad -m_{nn}$	-1	q_n	Δq_n	$q_n+\Delta q_n$

Let the vector q be changed to $\hat{q}=q+\Delta q$ where

$$\Delta q = \begin{bmatrix} \Delta q_1 \\ \vdots \\ \Delta q_s \\ \vdots \\ \Delta q_n \end{bmatrix}$$

In the initial tableau, the vector of changes Δq can be viewed as

$$\Delta q = \begin{bmatrix} 1 \ldots 0 \ldots 0 \\ \vdots \quad \vdots \quad \vdots \\ 0 \quad 1 \quad 0 \\ \vdots \quad \vdots \quad \vdots \\ 0 \quad 0 \quad 1 \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \vdots \\ \Delta q_s \\ \vdots \\ \Delta q_n \end{bmatrix}$$

i.e. We view as if the changes $\Delta q_1, \ldots, \Delta q_s, \ldots, \Delta q_n$ have been introduced as variables in the initial tableau with their corresponding activity vectors as $e_1, \ldots, e_s, \ldots, e_n$ though explicitly they need not be introduced in the tableau as the identity matrix itself is present in the initial tableu. If the same sequence of arithmetic operations of Lemke's pivot algorithm is carried throughout on the vector of changes Δq then it can be observed from the final tableau 2 that Δq transforms to Δq^* as below :

$$\Delta q^* = \begin{bmatrix} \Delta q_1^* \\ \vdots \\ \Delta q_s^* \\ \vdots \\ \Delta q_n^* \end{bmatrix} = \begin{bmatrix} \beta_{11}^* \cdots \beta_{1s}^* \cdots \beta_{1n}^* \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \beta_{s1}^* \quad \beta_{ss}^* \quad \beta_{sn}^* \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \beta_{n1}^* \quad \beta_{ns}^* \quad \beta_{nn}^* \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \vdots \\ \Delta q_s \\ \vdots \\ \Delta q_n \end{bmatrix}$$

$$\text{or} \qquad \Delta q_s^* = \sum_{j=1}^{n} \beta_{sj}^* \; \Delta q_j \;, \quad s = 1,\ldots,n. \qquad (2)$$

where $\{\beta_{sj}^*\}$ is a basis inverse present in the final LCP tableau 2.

(Tableau 2)

Basis	$w_1 \cdots w_s \cdots w_n$			$z_1 \cdots z_s \cdots z_n$			z_o	q^*	Δq^*	$\bar{q} = q^* + \Delta q^*$
$B.V_1$	β_{11}^*	β_{1s}^*	β_{1n}^*	m_{11}^*	m_{1s}^*	m_{1n}^*		q_1^*	Δq_1^*	$q_1^* + \Delta q_1^*$
$\vdots$							non-Basic			
$B.V_s$	β_{s1}^*	β_{ss}^*	β_{sn}^*	m_{s1}^*	m_{ss}^*	m_{sn}^*		q_s^*	Δq_s^*	$q_s^* + \Delta q_s^*$
$\vdots$										
$B.V_n$	β_{n1}^*	β_{ns}^*	β_{nn}^*	m_{n1}^*	m_{ns}^*	m_{nn}^*		q_n^*	Δq_n^*	$q_n^* + \Delta q_n^*$

and $\bar{q} = q^* + \Delta q^*$ which is same as obtained by performing the same sequence of pivot operations on $\hat{q}$. (Δq^* is easy to calculate than directly calculating $\bar{q}$ because many components of Δq may be zero.)

Thus, after solving LCP, when we find that there are some changes in elements of vector q, denoted by Δq, we calculate Δq^* by (2) and then $\bar{q} = q^* + \Delta q^*$. Now there are two possibilities :

<u>Case a</u> : $\bar{q} \geq 0$.

The LCP feasible basic vector w.r.t. q is also a LCP feasible basic vector w.r.t. $q + \Delta q$.

<u>Case b</u> : $\bar{q} \ngeq 0$ (i.e.for at least one i, $\bar{q}_i < 0$).

Continue the pivot operations of Lemke's method by taking current tableau as initial tableau for the changed problem till we get complimentary feasible solution.

<u>Parametrization of vector q</u>

Instead of considering discrete changes in q let q vary continuously as a linear function of some parameter α i.e.

$$\hat{q} = q + \alpha f$$

where f is some fixed vector, α is scalar, $\hat{q}$ being the changed vector of q. If $B^{-1} = \{\beta^*_{sj}\}$ is the basis inverse, corresponding to $\alpha = 0$, then

$$B^{-1}\hat{q} = B^{-1}(q + \alpha f)$$

$$= B^{-1}q + \alpha B^{-1}f$$

$$\bar{q} = q^* + \alpha B^{-1}f$$

$$\text{or} \qquad \bar{q}_s = q^*_s + \alpha \sum_{j=1}^{n} \beta^*_{sj} f_j$$

Select μ_1 and μ_2 such that

$$\mu_1 = \max_s \left\{ \frac{-q_s^*}{\sum_{j=1}^{n} \beta_{sj}^* f_j} \right\} \text{ for those } s \text{ for which } \sum_{j=1}^{n} \beta_{sj}^* f_j > 0$$

$$= -\infty \qquad \text{if for all } s \quad \sum_{j=1}^{n} \beta_{sj}^* f_j \leq 0$$

$$\mu_2 = \min_s \left\{ \frac{-q_s^*}{\sum_{j=1}^{n} \beta_{sj}^* f_j} \right\} \text{ for those } s \text{ for which } \sum_{j=1}^{n} \beta_{sj}^* f_j < 0$$

$$= \infty \qquad \text{if for all } s \quad \sum_{j=1}^{n} \beta_{sj}^* f_j \geq 0$$

For $\mu_1 \leq \alpha \leq \mu_2$, the final LCP feasible basis w.r.t. q is also a feasible basis w.r.t. $q+\alpha f$ which is the <u>case a</u> above. For $\alpha \notin [\mu_1, \mu_2]$, we have <u>case b</u>. The sensitivity analysis with regard to both cases has been studied above.

(ii) <u>Changes in columns of M</u>

For the sake of simplicity we first consider changes in one column of M. Suppose that the kth column of M is changed. Let the change be

$$\Delta m_k = \begin{bmatrix} \Delta m_{1k} \\ \vdots \\ \Delta m_{sk} \\ \vdots \\ \Delta m_{nk} \end{bmatrix}$$

This change, in the initial tableau can be viewed as follows :

$$\Delta m_k = \begin{bmatrix} 1 \ldots . 0 \ldots . 0 \\ \vdots \\ 0 & 1 & 0 \\ \vdots \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta m_{1k} \\ \Delta m_{sk} \\ \Delta m_{nk} \end{bmatrix}$$

i.e. it may be viewed as if $\Delta m_{1k}, \ldots \Delta m_{sk}, \ldots \Delta m_{nk}$ are variables with their corresponding activity vectors as $e_1, \ldots e_s, \ldots e_n$, though we really don't have to introduce them in the tableau as the identity matrix itself is present there.

By carrying out the same sequence of pivot operations of Lemke's method on the vector of changes Δm_k it can be observed from the final LCP tableau 2 that

$$\Delta m_k^* = \begin{bmatrix} \Delta m_{1k}^* \\ \vdots \\ \Delta m_{sk}^* \\ \vdots \\ \Delta m_{nk}^* \end{bmatrix} = \begin{bmatrix} \beta_{11}^* \ldots \beta_{1s}^* \ldots \beta_{1n}^* \\ \beta_{s1}^* & \beta_{ss}^* & \beta_{sn}^* \\ \beta_{n1}^* & \beta_{ns}^* & \beta_{nn}^* \end{bmatrix} \begin{bmatrix} \Delta m_{1k} \\ \Delta m_{sk} \\ \Delta m_{nk} \end{bmatrix}$$

$$\text{or} \quad \Delta m_{sk}^* = \sum_{j=1}^{n} \beta_{sj}^* \, \Delta m_{jk} \, , \quad s = 1, 2, \ldots, n \qquad (3)$$

$$\text{and} \quad \bar{m}_k = m_k^* + \Delta m_k^*$$

After affecting this change in final LCP tableau, we have two possibilities :

Case a : If kth column is such that the corresponding variable is a non-basic variable then the final complimentary basic feasible solution w.r.t. old matrix continues to be complementary basic feasible solution w.r.t. changed matrix. In fact in such a case there is no need to calculate $\bar{m}_k$.

Case b : Now suppose that the kth column is such that the corresponding variable is a basic variable in the final LCP tableau. If kth column is in t-th place, say, of the basis then before changes

$$m^*_{tk} = 1$$

$$m^*_{ik} = 0, \; i \neq t$$

After changes, the kth column looks like

$$\bar{m}_k = \begin{bmatrix} \Delta m^*_{1k} \\ \cdot \\ \cdot \\ 1 + \Delta m^*_{tk} \\ \cdot \\ \cdot \\ \Delta m^*_{nk} \end{bmatrix}$$

This effects the basic structure of the tableau. To reduce the tableau to its proper form, we pivot the current tableau with respect to (t,k)th element where by a proper form we mean the form of the tableau where a basic vector has a unit element corresponding to its place of appearance in the basis and zeroes elsewhere. However, we have assumed above

that we can always pivot (t,k)th element i.e. we have assumed that

$$m^*_{tk} \neq -1.$$

The results of pivot operations on q^* are

$$\bar{q}_i = q^*_i - \frac{\Delta m^*_{ik} q^*_t}{(1+ \Delta m^*_{tk})}$$

$$\bar{q}_t = \frac{q^*_t}{(1+ \Delta m^*_{tk})}$$

where $\bar{q}$ is the transformed vector. If $\bar{q} \geq 0$, the current LCP basis continues to be complementary feasible basis. If $\bar{q} \ngeq 0$, restart the pivot operations of Lemke's method taking final tableau as initial tableau for the changed problem till we get final complimentary basic feasible vector.

Changes in more than one column of M can be treated the same way. First we affect all the changes in the LCP tableau and then transform the tableau to its proper form. Finally we deal with the transformed q.

(iii) Simultaneous changes in M and q

Changes in columns of M and vector q can be considered simultaneously as well. At first, changes with respect to all columns of M and column q should be affected

together in the final LCP tableau. The pivot operations are applied to deal with the disturbed tableau to restore it in the proper form corresponding to the positions of its basis vectors and then finally treat the q vector. We again assume here that pivoting is possible at each step.

3.3 Illustration

Sensitivity Analysis of Quadratic Programming Problem

Boot [27] has considered the sensitivity analysis of convex quadratic programs by considering infinitesimal small changes in the parameters of the problem. However the results are approximately valid for small changes of the problem parameters. We consider here the post-optimization of quadratic programming problems by first converting them to LCP. The analysis takes care of all types of changes, small or large. The procedure is illustrated below :

Example 1 : Consider the convex quadratic program

$$\text{(I)} \quad \begin{aligned} \min \quad & f(x) = -6x_1+2x_1^2-2x_1x_2+2x_2^2 \\ \text{subject to} \quad & -x_1-x_2 \geq -2 \\ & x_1 \geq 0, \; x_2 \geq 0 \end{aligned}$$

Viewing the objective function and the constraints in the standard form

$$\min \quad f(x) = cx + \frac{1}{2} x^T G x$$

subject to

$$Ax \geq b$$

$$x_1, x_2 \geq 0$$

the corresponding LCP is derived below :

$$M = \begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix} = \begin{bmatrix} -6 \\ 0 \\ 2 \end{bmatrix}$$

and LCP is $w - Mz = q$

The tabular form for which is (tableau 3)

(Tableau 3)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	z_o	q	Δq
w_1	1	0	0	-4	2	-1	-1	-6	-4
w_2	0	1	0	2	-4	-1	-1	0	6
w_3	0	0	1	1	1	0	-1	2	4

(Δq column shown here for convenient future calculations)

Solving by Lemke's Complementary Pivot Method [119], the final LCP tableau is (tableau 4)

(Tableau 4)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	z_o	q^*	Δq^*	$\bar{q} = q^* + \Delta q^*$
z_3	$-\frac{1}{2}$	$-\frac{1}{2}$	-1	0	0	1	-1	1	-5	-4
z_1	$-\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{2}$	1	0	0	-1	$\frac{3}{2}$	$\frac{17}{6}$	$\frac{13}{3}$
z_2	$\frac{1}{12}$	$-\frac{1}{12}$	$\frac{1}{2}$	0	1	0	-1	$\frac{1}{2}$	$\frac{7}{6}$	$\frac{5}{3}$

(The column z_o is not the result of calculations)

The optimal solution to primal problem (I) is $x_1 = \frac{3}{2}$, $x_2 = \frac{1}{2}$

Change in q : Suppose as a result of some policy decisions the quadratic program is changed as

$$\min \quad f(x) = -10x_1+6x_2+2x_1^2-2x_1x_2+2x_2^2$$

(II) subject to

$$-x_1-x_2 \geq -6$$
$$x_1, x_2 \geq 0$$

Here M remains the same and

$$q + \Delta q = \begin{bmatrix} -10 \\ 6 \\ 6 \end{bmatrix} \quad \text{so that} \quad \Delta q = \begin{bmatrix} -4 \\ 6 \\ 4 \end{bmatrix}$$

To avoid space and for the sake of computational convenience, Δq column is listed in tableau 3 as shown. This is particularily

convenient if the LCP solutions are needed for set of values of Δq's because in that case pivot operations may simultaneously be carried over these changes. We have

$$\Delta q^* = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -1 \\ -\frac{1}{12} & \frac{1}{12} & \frac{1}{2} \\ \frac{1}{12} & -\frac{1}{12} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -4 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} -5 \\ \frac{17}{6} \\ \frac{7}{6} \end{bmatrix}$$

$$\bar{q} = q^* + \Delta q^* = \begin{bmatrix} -4 \\ \frac{13}{3} \\ \frac{5}{3} \end{bmatrix}$$

Regarding this final LCP tableau (4) as initial tableau and applying Lemke's procedure to affected tableau we reach the following tableau 5.

(Tableau 5)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	z_o	$\dot{\bar{q}}$
w_2	$\frac{1}{2}$	1	0	0	-3	$-\frac{3}{2}$		1
z_1	$-\frac{1}{4}$	0	0	1	$-\frac{1}{2}$	$\frac{1}{4}$	Non-Basic	$\frac{5}{2}$
w_3	$\frac{1}{4}$	0	1	0	$\frac{3}{2}$	$-\frac{1}{4}$		$\frac{7}{2}$

Optimal solution to changed primal problem (II) is
$x_1^* = \frac{5}{2}$, $x_2^* = 0$, $f(x^*) = -\frac{25}{2}$.

It took three pivot operations by post-optimization procedure to solve the changed problem and two steps when the problem was solved all over again with the changed data. However, in this case, the change Δq was a very large change with respect to q.

Change in M and q

Let us now consider changes in elements of M and q both after we have solved program (I).

Consider the changed quadratic program

$$\min \quad f(x) = -6x_1+2x_1^2-2x_1x_2+2x_2^2$$

(III) subject to

$$-x_1-2x_2 \geq -4$$

$$x_1, x_2 \geq 0$$

$$-M+\Delta M = -\begin{bmatrix} 4 & -2 & 1 \\ -2 & 4 & 2 \\ -1 & -2 & 0 \end{bmatrix}, \quad q+\Delta q = \begin{bmatrix} -6 \\ 0 \\ 4 \end{bmatrix}$$

Therefore

$$\Delta m_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \Delta m_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \Delta m_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \Delta q = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Again for the sake of computational convenience, the changes Δq, Δm_2 and Δm_3 may be introduced in the initial tableau. We, however, give below the final LCP tableau (6) (with regard to matrix M and vector q) with changes introduced as below.

(Tableau 6)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	q^*	Δq^*	Δm_2^*	Δm_3^*
z_3	$-\frac{1}{2}$	$-\frac{1}{2}$	-1	0	0	1	1	-2	-1	$\frac{1}{2}$
z_1	$-\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{2}$	1	0	0	$\frac{3}{2}$	1	$\frac{1}{2}$	$-\frac{1}{12}$
z_2	$\frac{1}{12}$	$-\frac{1}{12}$	$\frac{1}{2}$	0	1	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{12}$

After affecting these changes we get tableau 7

(Tableau 7)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	$\bar{q} = q^* + \Delta q^*$
z_3	$-\frac{1}{2}$	$-\frac{1}{2}$	-1	0	-1	$\frac{3}{2}$	-1
z_1	$-\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$-\frac{1}{12}$	$\frac{5}{2}$
z_2	$\frac{1}{12}$	$-\frac{1}{12}$	$\frac{1}{2}$	0	$\frac{3}{2}$	$\frac{1}{12}$	$\frac{3}{2}$

Basic vectors in columns corresponding to z_2 and z_3 are not appearing in their proper form. To bring them in the proper form, we pivot around the elements (3,5) and (1,6) and reach tableau 8.

(Tableau 8)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	$\bar{q}$
z_3	$-\frac{2}{7}$	$-\frac{5}{14}$	$-\frac{3}{7}$	0	0	1	0
z_1	$-\frac{1}{7}$	$\frac{1}{14}$	$\frac{2}{7}$	1	0	0	2
z_2	$\frac{1}{14}$	$-\frac{1}{28}$	$\frac{5}{14}$	0	1	0	1

Since $\bar{q} \geq 0$, the final LCP tableau is reached.

Optimal solution to primal problem (III) is $x_1 = 2$, $x_2 = 1$ with optimal value = -6.

The post-optimization procedure gave the solution to changed quadratic program (III) in just two pivot operations, needed to restore the tableau to its proper form. When the changed program (III) was solved all over again, it took three pivot operations (Sometimes it is useful in sensitivity analysis to know the set of values for model parameters for which the solution remains optimal).

Convenient Changes in Quadratic Programs

The LCP formulation of a quadratic program indicates that changes in b, c and a column of G can be conveniently carried out by the method described above. However, changes in A are more cumbersome, in that if one column of A is changed, then many columns of the M matrix are affected. Similarly, the LCP formulation of a LP indicates that changes in the cost vector c and requirement vector b can be conveniently carried out while the changes in A are cumbersome.

3.4 Applications

3.4.1 Linear Fractional Functional Programming

Linear Fractional Functional Program (LFFP) is

$$\min \quad f(x) = \frac{c^T x+\alpha}{d^T x+\beta}$$

(LFFP) subject to

$$Ax \geq b$$

$$x \geq 0$$

with the condition that $d^T x+\beta > 0$ for all feasible x. Charnes and Cooper [31], Dinkelbach [44] and Martos [114] have given methods for LFFP. We use the technique of Dinkelbach [44] to solve the LFFP. The parameterised

LPs required to be solved are handled by considering the sensitivity analysis of q in the corresponding LCP. By Dinkelbach's technique we have to solve

$$\min \quad \varphi(x,z) = (c^T x+\alpha) - z(d^T x+\beta)$$

$$= (c-zd)^T x + (\alpha - \beta z)$$

(PLP) $\quad$ subject to $\quad Ax \geq b$

$$x \geq 0$$

for different values of z till $\varphi(x,z) = 0$.

The corresponding LCP is

$$w - Mz = q$$

$$\text{where } q = \begin{bmatrix} (c-zd) \\ -b \end{bmatrix}$$

We denote this by LCP(q).

<u>Algorithm</u> : The algorithm is described in the following steps.

<u>Step 1</u> : Set $z = z^0 = 0$ and solve LCP(q_o) with $q_o = \begin{bmatrix} c \\ -b \end{bmatrix}$. Let x^0 be optimal solution to corresponding primal problem (PP). Set $i = 0$.

<u>Step 2</u> : (i) If $\varphi(x^i,z^i) = 0$ then x^i is the optimal solution to (LFFP). Stop.

(ii) If $\varphi(x^i,z^i) < \delta$, where δ is some pre-assigned

positive number then x^i is the approximate solution to (LFFP). Stop. Otherwise, compute.

$$z^{i+1} = \frac{c^T x^i + \alpha}{d^T x^i + \beta}$$

and go to step 3 with $i = i+1$.

Step 3 : Solve LCP(q_i) for

$$q_i = \begin{bmatrix} (c - z^i d) \\ -b \end{bmatrix}$$

by sensitivity analysis applied to LCP(q_{i-1}) taking vector of changes $\Delta q_{i-1} = q_i - q_{i-1}$. Let x^i be the optimal solution to corresponding primal problem. go to step 2.

Example 2

Below we illustrate the above procedure for (LFFP) by an example.

$$\min \quad f(x) = \frac{x_1 + x_2}{2x_1 + 3x_2 + 1}$$

subject to

$$\text{(LFFP)} \qquad -x_1 - x_2 \geq -2$$

$$x_1 + x_2 \geq 1$$

$$x_1, x_2 \geq 0$$

Let S be the feasible set.

Iteration 0

Step 1 : Set $z = z^{o} = 0$

$$\min \quad \varphi(x,z^{o}) = x_1+x_2$$

(PLP_o)

subject to $x \in S$

Corresponding LCP is $w-Mz = q_o$, where

$$M = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, q_o = \begin{bmatrix} 1 \\ 1 \\ 2 \\ -1 \end{bmatrix}$$

The final LCP tableau (9) for LCP(q_o) is

(Tableau 9)

Basis	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	q_o^*	Δq_o^*	$q_1^*=q_o^*+\Delta q_o^*$
w_1	1	-1	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$
z_4	0	1	0	0	0	0	-1	1	1	$-\frac{3}{4}$	$\frac{1}{4}$
w_3	0	0	1	1	0	0	0	0	1	0	1
z_2	0	0	0	-1	1	1	0	0	1	0	1

Optimal solution to (PLP_o) is $x^o = (z_1, z_2) = (o, 1)$.

Step 2 : $\varphi(x^o, z^o) \neq 0$, $z^1 = \frac{1}{4}$

Iteration 1

Step 3 : Solve $\min \varphi(x, z^1) = \frac{x_1}{2} + \frac{x_2}{4} - \frac{1}{4}$

(PLP_1) subject to $x \in S$

$$q_1 = q_o + \Delta q_o = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{4} \\ 2 \\ -1 \end{bmatrix}, \quad \text{Therefore } \Delta q_o = \begin{bmatrix} -\frac{1}{2} \\ -\frac{3}{4} \\ 0 \\ 0 \end{bmatrix}$$

$$\Delta q_o^* = \begin{bmatrix} \frac{1}{4} \\ -\frac{3}{4} \\ 0 \\ 0 \end{bmatrix}, \quad q_1^* = q_o^* + \Delta q_o^* = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ 1 \\ 1 \end{bmatrix}$$

Since $q_1^* \geq 0$, LCP tableau (9) continues to be final LCP tableau for LCP(q_1) also. Optimal solution to (PLP_1) is $x^1 = (0, 1)$.

Step 2 : $\varphi(x^1, z^1) = 0$. Therefore $x^1 = (0, 1)$ solves (LFFP) and its optimal value is $= \frac{1}{4}$.

3.4.2 Quadratic Fractional Functional Programming

Quadratic Fractional Functional Program (QFFP) is

$$\text{(QFFP)} \quad \min \frac{\frac{1}{2} x^T G x + c^T x + \alpha}{d^T x + \beta}$$

subject to

$$Ax \geq b$$

$$x \geq 0$$

with the conditions that $d^T x+\beta > 0$ for all feasible x and G is positive semi-definite (PSD) matrix. We use the technique of Dinkelbach to solve QFFP. The parametrized quadratic programs are handled by considering the sensitivity analysis of q in the corresponding LCP. By Dikelbach's method we have to solve

$$\min \quad \varphi(x,z) = \frac{1}{2} x^T G x + (c-zd)^T x + (\alpha-\beta z)$$

subject to

$$\text{(PQP)} \qquad Ax \geq b$$

$$x \geq 0$$

for different values of z till $\varphi(x,z) = 0$. The corresponding LCP is w-Mz = q where

$$q = \begin{bmatrix} (c-zd)^T \\ -b \end{bmatrix}$$

We will denote this by LCP(q).

The steps of the algorithm to solve this program are essentially the same as described in section 3.4.2.

Example 3

In the following we illustrate the procedure by an example.

$$\min \quad f(x) = \frac{2x_1^2-2x_1x_2+2x_2^2-6x_1}{x_1+x_2+1}$$

(QFFP)

subject to

$$-x_1-x_2 \geq -2$$

$$x_1, x_2 \geq 0$$

Let S be the feasible set of the problem.

Iteration 0

Step 1 : Set $z = z^o = 0$

$$\min \quad \varphi(x, z^o) = 2x_1^2-2x_1x_2+2x_2^2-6x_1$$

(PQP_o)

subject to $x \in S$

$$M = \begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad q_o = \begin{bmatrix} -6 \\ 0 \\ 2 \end{bmatrix}$$

Solve LCP(q_o) which is the same as that of convex quadratic program of section 3.3 and the tableau 4 is the final LCP tableau.

Optimal solution to (PQP$_o$) is $x^o = (\frac{3}{2}, \frac{1}{2})$.

Step 2 : $\varphi(x^o, z^o) = -\frac{11}{2} \neq 0, \quad z^1 = -\frac{11}{6}$

Iteration 1

Step 3 : Solve $\quad \min \ \varphi(x, z^1) = 2x_1^2 - 2x_1x_2 + 2x_2^2 + (-\frac{25}{6}x_1 + \frac{11}{6}x_2) + \frac{11}{6}$

(PQP$_1$) $\qquad$ subject to $\quad x \in S$

$$q_1 = q_o + \Delta q_o = \begin{bmatrix} -\frac{25}{6} \\ \frac{11}{6} \\ 2 \end{bmatrix}, \quad \text{Therefore} \quad \Delta q_o = \begin{bmatrix} \frac{11}{6} \\ \frac{11}{6} \\ 0 \end{bmatrix}$$

$$\Delta q_o^* = \begin{bmatrix} -\frac{11}{6} \\ 0 \\ 0 \end{bmatrix}, \quad q_1^* = q_o^* + \Delta q_o^* = \begin{bmatrix} -\frac{5}{6} \\ \frac{3}{2} \\ \frac{1}{2} \end{bmatrix}$$

Solve LCP(q_1) by applying sensitivity analysis procedure to final LCP tableau of LCP(q_o). The final tableau for LCP(q_1) is

(Tableau 10)

Basis	w_1	w_2	w_3	z_1	z_2	z_3	q_1^*	Δq_1^*	$q_2^* = q_1^* + \Delta q_1^*$
w_3	$\frac{1}{2}$	$\frac{1}{2}$	1	0	0	-1	$\frac{5}{6}$	$\frac{1}{6}$	1
z_1	$-\frac{1}{3}$	$-\frac{1}{6}$	0	1	0	$\frac{1}{2}$	$\frac{13}{12}$	$-\frac{1}{12}$	1
z_2	$-\frac{1}{6}$	$-\frac{1}{3}$	0	0	1	$\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{12}$	0

(Notation : q_i^*, after the result of pivot operations leading to final LCP tableau, is again referred as q_i^* though it is its

Optimal solution to (PLP$_1$) is $x^1 = (\frac{13}{12}, \frac{1}{12})$.

Step 2 : $\varphi(x^1, z^1) = -\frac{25}{72} \neq 0$. Compute $z^2 = -\frac{311}{156} \simeq -2$

Iteration 2

Step 3 : Solve min $\varphi(x, z^2) = (2x_1^2 - 2x_1x_2 + 2x_2^2) + (-4x_1 + 2x_2) + 2$

(PQP$_2$) subject to $x \in S$

$$q_2 = q_1 + \Delta q_1 = \begin{bmatrix} -4 \\ 2 \\ 2 \end{bmatrix}, \text{ Therefore } \quad \Delta q_1 = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ 0 \end{bmatrix}$$

$$\Delta q_1^* = \begin{bmatrix} \frac{1}{6} \\ -\frac{1}{12} \\ -\frac{1}{12} \end{bmatrix}, \quad q_2^* = q_1^* + \Delta q^* = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Since $q_2^* \geq 0$, tableau 10 is final LCP tableau for LCP(q_2). The optimal solution to (PLP$_2$) is $x^2 = (1,0)$.

Step 2 : $\varphi(x^2, z^2) = 0$. Therefore $x^2 = (1,0)$ solves (QFFP) and its optimal value is $= -2$.

3.4.3 Another Quadratic Fractional Functional Program

Consider another Quadratic Fractional Functional Program

$$\min f(x) = \frac{\frac{1}{2}x^T Gx + c^T x + \alpha}{\frac{1}{2}x^T Hx + d^T x + \beta}$$

subject to

$$Ax \geq b$$

$$x \geq 0$$

with the conditions that numerator ≥ 0, denominator > 0 for every feasible x, G is PSD and H is ND. The problem can be solved by sensitivity analysis of LCP considering changes in q only which are given by parameterization technique of Dinkelbach. We have to solve

$$\min \quad \varphi(x, z) = \frac{1}{2}x^T (G - zH)x + (c - zd)^T x + (\alpha - \beta z)$$

subject to

$$Ax \geq b$$

$$x \geq 0$$

for different values of z till $\varphi(x, z) = 0$.

Conclusion

We have demonstrated that sensitivity analysis of all those problems which can be formulated as LCP can be conveniently discussed through sensitivity analysis of LCP.

Furthermore, some of the problems find their easy solutions through changed LCP. A simple computational scheme for such problems has been presented.

CHAPTER - 4

RELAXATION APPROACH FOR BOUNDED VARIABLE LINEAR PROGRAMS*

4.1 Introduction

The Bounded Variable Linear Program (BVLP) is

$$\max \quad cx$$

subject to

$$P(J) \qquad \begin{aligned} Ax &= b \\ x &\geq 0 \\ 0 \leq x_j &\leq u_j, \; j \in J \end{aligned}$$

where A is an mxn matrix, b is m-vector, $x \in R^n$, $J \subset I_n = \{1,2,\ldots,n\}$, the index set of variables. Any linear program (LP) can also be viewed as BVLP taking bounds on x_j to be very large.

Though the problem can be solved by carryingout an m+n equation system , it becomes computationally laborious. Dantzig [40] has proposed a method to solve the BVLP by restricting the basis size to m only. Refer to [76,119,148] for good expositions of this method. This method has the

*The work contained in this chapter was presented in Annual Meeting of Bharat Ganit Parishad held at Lucknow (India) in April,1979 [90,91].

following drawbacks :

(i) Though the criteria for the selection of the vector to enter the basis is the same as in simplex method (i.e. the vector a_j associated with most negative z_j-c_j enters the basis), the criteria for deciding which vector should leave the basis is pretty involved.

(ii) There exist many physical situations where the optimal LP solution (without bounds) violates only a few upper bound constraints. The existing method does not give us any information about this. (From here onwards LP without bounds is simply referred as LP).

We propose two methods to solve the BVLP which take care of the above drawbacks, to a certain extent. The proposed methods essentially fall under the relaxation approach, i.e. we relax the upper bound constraints and then introduce them sequentially. The first method makes use of the Dantzig's method with one upper bound constraint only and the second method is completely independent of it.

In section 4.2 we prove a theorem which contributes to the development of the proposed methods. In section 4.3, a method making use of Dantzig's method is given. In section 4.4, a dual simplex approach is proposed to solve the relaxed problem.

4.2 Main Result

The BVLP with index set J for the bounded variables will be denoted by P(J). We prove a theorem below which contributes to the development of the methods proposed in the following sections.

Theorem : If the optimal solution to $P(\varphi)$ violates only one of the upper bound constraints, say, for $j = p$, then in every optimal solution to P(J), $x_p = u_p$.

Proof : Let $x^o = (x_1^o, x_2^o, \ldots, x_n^o)$ be the optimal solution of $P(\varphi)$ such that $x_p^o > u_p$. Let S(J) be the constraint set of P(J).

Thus $S(\{p\}) = \left\{x \mid Ax = b,\ x \geq 0,\ x_p \leq u_p\right\}$.

For $t \leq u_p$, define $S(p,t) = \left\{x \mid Ax = b,\ x \geq 0,\ x_p \leq t\right\}$.

Clearly $S(p,t) \subseteq S(p)$ and $S(p,u_p) \equiv S(p)$.

$$\text{Let} \quad z_{max}(t) = \max_{x \in S(p,t)} cx.$$

Since $S(p,t_1) \subseteq S(p,t_2)$ for $t_1 < t_2 \leq u_p$, it follows that $z_{max}(t)$ is an increasing function of t in the interval $0 \leq t \leq u_p$. Also because, $x_p^o > u_p$, $z_{max}(t)$ is a strictly increasing function. Hence $x_p = u_p$ in every optimal solution of P(J).

4.3 The Artificial Variable Technique

In this section, we describe an iterative technique

to solve P(J) starting with $P(\varphi)$. Let P(I) denote the BVLP

$$\max \ cx$$

subject to

P(I)

$$Ax = b$$

$$0 \le x_j \le u_j \quad \forall \ j \in I \subset J$$

$$x \ge 0$$

Let $x^o(I)$ denote the optimal solution of P(I) and $x_j^o(I)$ denote the jth component of $x^o(I)$. If $x_j^o(I) \le u_j \ \forall \ j \in J$, then $x^o(I)$ solves P(J). Otherwise there exists a $p \in J-I$ such that $x_p^o(I) > u_p$. The essential step of the iterative scheme is to solve the BVLP

$$\max \ cx$$

subject to

$P(I \cup \{p\})$

$$Ax = b$$

$$0 \le x_j \le u_j \quad \forall \ j \in I \cup \{p\}$$

$$x \ge 0$$

starting with the solution of P(I), which is achieved through the following steps referred to as Substitution Steps.

Substitution Steps

Substitute x_p at its upper bound u_p (Previous Theorem). Introduce an artificial variable* x_{pa} corresponding to x_p with $-M$ price, M being very large positive number at a level $x_{pa} = x_p^o - u_p$ and consider it a basic variable in place of x_p. The column of y_{ij}'s under the artificial variable x_{pa} would be the same as that of x_p.

Recalculate new $(z_j - c_j)$'s row and the objective function value z.

$$\hat{z} = z + (x_p^o - u_p)\ (-c_p) + (x_p^o - u_p)\ (-M)$$

$$= z - (x_p^o - u_p)\ (M + c_p)$$

If x_p was present in ith place of basis (the place taken by x_{pa} now) then

$$\hat{z}_j - c_j = (z_j - c_j) - My_{ij} - c_p y_{ij}$$

$$= (z_j - c_j) - (M + c_p) y_{ij}, \text{ for all } j$$

$$z_{pa} - c_{pa} = 0 , \quad \text{for the artificial variable.}$$

Thus $z_p - c_p = -(M + c_p) < 0$.

Replace x_p by $-x_p'$ where $0 \leq x_p' \leq u_p$ (The variables x_p and x_p' are

*We will confine our discussion to original simplex method [76] only.

$$x_p^o - u_p = \max_i \; (x_i^o - u_i) \qquad (1)$$

Step 4 : Obtain $x^o(I \cup \{p\})$. We know by the theorem proved in section 4.2 that $x_p^o(I \cup \{p\}) = u_p$.

Set $I = \{j \in J \mid x_j^o(I \cup \{p\}) \le u_j\}$.

Step 5 : If $I = J$, $x^o(I \cup \{p\}) = x^o(J)$ otherwise go to step 3.

Convergence

If in the optimal solution to mxn linear program without bounds, r basic variables are within their bounds, then at the most (in worst case) m-r iterations are needed. Thus the method surely converges. It may become computationally laborious in cases where a large number of variables in solution $x^o(\varphi)$ exceed their upper bounds. However, it is conceivable that $|J|$ is large, but $|J-I|$, with I in step 1, is small. In such situations, the technique will be useful.

Illustration

The method is illustrated below by an example.

Example max $z = x_1+4x_2+3x_3+x_4+\frac{3}{2}x_5$

subject to

$$x_1+2x_2+x_3+x_4+x_5 \le 24$$

$$2x_1+x_2+4x_3+x_4 \le 48$$

$$x_1+3x_2+2x_3+3x_4+x_5 \le 36$$

$0 \le x_1 \le 8$, $0 \le x_2 \le 6$, $0 \le x_3 \le 8$, $0 \le x_4 \le 4$, $0 \le x_5 \le 10$.

Iteration 1

Step 1 : The optimal simplex **tableau is given** below.

(Tableau 1)

$c_j \rightarrow$		1	4	3	1	3/2	0	0	0
c_B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$c_5=\frac{3}{2}$	$x_5=12$	$\frac{5}{3}$	0	0	$-\frac{10}{3}$	1	$\frac{10}{3}$	$\frac{1}{3}$	$-\frac{7}{3}$
$c_3=3$	$x_3=12$	$\frac{2}{3}$	0	1	$-\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$
$c_2=4$	$x_2=0$	$-\frac{2}{3}$	1	0	$\frac{7}{3}$	0	$-\frac{4}{3}$	$-\frac{1}{3}$	$\frac{4}{3}$
	54	$\frac{5}{6}$	0	0	$\frac{7}{3}$	0	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{5}{6}$

Step 2 : $I = \{1,2,4\} \subset J = \{1,2,3,4,5\}$

Step 3 : $p = 3$, by (1)

Step 4 : Substitution steps give the following tableau.

(Tableau 2)

$c_j \rightarrow$		1	4	-3	1	3/2	0	0	0	$-M$
c_B	x_B	x_1	x_2	x_3'	x_4	x_5	x_6	x_7	x_8	x_{3a}
$c_5=\frac{3}{2}$	$x_5=12$	$\frac{5}{3}$	0	0	$-\frac{10}{3}$	1	$\frac{10}{3}$	$\frac{1}{3}$	$-\frac{7}{3}$	0
$c_{3a}=-M$	$x_{3a}=4$	$\boxed{\frac{2}{3}}$	0	-1	$-\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	1
$c_2 = 4$	$x_2=0$	$-\frac{2}{3}$	1	0	$\frac{7}{3}$	0	$-\frac{4}{3}$	$-\frac{1}{3}$	$\frac{4}{3}$	0
	$42-4M$	$(-\frac{2M}{3}-\frac{7}{6})$,	0	$(M+3)$	$(\frac{M}{3}+\frac{10}{3})$,	0,	$(-\frac{M}{3}-\frac{1}{3})$,	$(-\frac{M}{3}-\frac{5}{6})$,	$(\frac{M}{3}+\frac{11}{6})$,	0

(Double bar vertical lines are put to distinguish the column p = 3 from other columns).

The problem is treated as BVLP in x_1, x_2, x_3, x_4' variables only (The notation used is of Taha [148]).

$$\theta_1 = 6 \quad \theta_2 = 9, \quad u_1 = 8$$

$$\theta = \min(6,9,8) = 6$$

We get the tableau 3 (after carrying out usual simplex iteration) as

(Tableau 3)

$c_j \rightarrow$		1	4	-3	1	3/2	0	0	0
c_B	x_B	x_1	x_2	x_3'	x_4	x_5	x_6	x_7	x_8
$\frac{3}{2}$	$x_5=2$	0	0	$\frac{5}{2}$	$-\frac{5}{2}$	1	$\frac{5}{2}$	$-\frac{1}{2}$	$-\frac{3}{2}$
1	$x_1=6$	1	0	$-\frac{3}{2}$	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\boxed{\frac{1}{2}}$	$-\frac{1}{2}$
4	$x_2=4$	0	1	-1	2	0	-1	0	1
	49	0	0	$\frac{5}{4}$	$\frac{11}{4}$	0	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{5}{4}$

$$\theta_1 = 12, \ \theta_2 = \infty, \ u_7 = \infty, \ \theta = 12$$

The next tableau is

(Tableau 4)

c_j	$\rightarrow$	1	4	-3	1	3/2	0	0	0
c_B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	$x_5=8$	1	0	1	-3	1	3	0	-2
	$x_7=12$	2	0	-3	-1	0	1	1	-1
	$x_2=4$	0	1	-1	2	0	-1	0	1
	52	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{5}{2}$	0	$\frac{1}{2}$	0	1

Optimality is reached.

Step 5 : All the variables are within their bounds.

So optimal solution is $x_2 = 4, x_3=8, x_5=8, x_1=x_4=0, z=52.$

4.4 Relaxed Dual Simplex Approach

In this section, we introduce a method to solve the BVLP which is completely independent of the Dantzig's method, unlike the one proposed in section 4.3. The method relaxes both the lower and upper bounds, as and when required, and the dual simplex algorithm is used at all the iterations, except the first iteration where simplex algorithm is used.

Let $\mathbf{x}^o$ be an optimal solution of the LP. If $p \in J$ and $x_p^o > u_p$, then we give below a procedure which introduces the constraint $x_p \leq u_p$.

The constraint $0 \le x_p \le u_p$ is equivalent to $x_p + x'_p = u_p$ and $0 \le x'_p \le u_p$. Relaxing the upper bound on x'_p is equivalent to relaxing the lower bound on x_p and vice versa.

If $x_p^o = x_{B_i}$ in the final simplex tableau of LP then

$$x_{B_i} = x_p + \sum_{j \in R} y_{ij} \, x_j$$

where R is the index set of non-basic variables. Using $x_p + x'_p = u_p$, in the above equation, we get

$$u_p - x_{B_i} = x'_p + \sum_{j \in R} (-y_{ij}) x_j$$

where $u_p - x_{B_i} = u_p - x_p^o < 0$.

Thus $x_p \le u_p$ can be incorporated in the tableau by affecting the following changes :

(i) replacing x_p^o by $u_p - x_p^o$

(ii) replacing variable x_p by x'_p

(iii) changing c_p to $-c_p$

(iv) changing sign of all elements in the ith row except the sign of "1" in (i,p)th position.

We can observe that sign of $(z_j - c_j)$'s remains unchanged. The above steps are referred to as Substitution Steps.

The resulting problem can be solved by using the dual simplex algorithm. The solution obtained will be of the problem

$$\max cx$$

(P) subject to

$$Ax = b$$
$$x_p \leq u_p$$
$$x_j \geq 0 \quad , \quad j \neq p.$$

We observe that $z_2 \leq z_1$ (by dual simplex iteration) where z_1 is the optimal value of LP and z_2 is the optimal value of the program (P).

The variable x_p' is treated now onwards as a bounded variable with upper bound u_p in order to impose the lower bound on x_p.

In general, at any stage of the algorithm, the relaxed problem being considered is

$$\max cx$$

subject to

$$Ax = b$$
$$x_j \leq u_j \quad , \quad j \in J_1$$
$$x_j \geq 0 \quad , \quad j \in I_n - J_1$$

Since this problem is a relaxation of the original BVLP, we get

$$z_B \leq z_R$$

where z_B is the optimal value of the original BVLP and z_R is the optimal value of the above relaxed problem.

If the solution to the above relaxed problem is not feasible for the original BVLP, then one of the variables (x_k or x_k') violates its upper bound. Such a variable is treated by the substitution steps and we get a new relaxed problem.

This new relaxed problem is one of the following :

$\underline{k \in J-J_1}$

$$\max \quad cx$$

(I) subject to

$$Ax = b$$

$$x_j \leq u_j, \quad j \in J_1 \cup \{k\}$$

$$x_j \geq 0, \quad j \in I_n - (J_1 \cup \{k\}).$$

$\underline{k \in J_1}$

$$\max \quad cx$$

(II) subject to

$$Ax = b$$

$$x_j \leq u_j \ , \quad j \in J_1 - \{k\}$$

$$x_j \geq 0 \ , \quad j \in I_n - \ (J_1 - \{k\}\ )$$

In either case,

$$z_R' \leq z_R$$

where z_R' is the optimal value of the new relaxed problem. The algorithm can now be be described as below :

Algorithm

Step 1 : Solve the LPP without bounds by simplex method.

Step 2 : If all the variables are within their bounds, we have solved the BVLP. Otherwise choose p as in (1).

Step 3 : Perform the substitution steps.

Step 4 : Apply dual simplex method and return to step 2.

Since value of the objective function at each step is decreasing and the upper bound constraints $x_j \leq u_j$, $j \in J$ which are to be incorporated in the tableau are finite, the method converges in finite number of steps.

Finally we point out that the algorithm has some drawbacks also. (i) It will take at least one pivot operation of simplex method to remove the infeasibility introduced by the substitution steps. (ii) If LPP without bounds is an unbounded problem then the algorithm may fail to find the solution because of its failure at the first step even though the feasible region of BVLP is bounded.

<u>Illustration</u>

The method is illustrated by a numerical example.

<u>Example</u> max $z = 3x_1+2x_2$

subject to

$$5x_1+2x_2 \leq 70$$

$$-2x_1+5x_2 \leq 30$$

$$0 \leq x_1 \leq 7$$

$$0 \leq x_2 \leq 7$$

<u>Iteration 0</u>

<u>Step 1</u> : The optimal simplex tableau (5) by simplex method [76] is given below :

(Tableau 5)

$c_j \rightarrow$		3	2	0	0
c_B	x_B	x_1	x_2	x_3	x_4
3	$x_1=10$	1	0	$\frac{5}{29}$	$-\frac{2}{29}$
2	$x_2=10$	0	1	$\frac{2}{29}$	$\frac{5}{29}$
	50	0	0	$\frac{19}{29}$	$\frac{4}{29}$

Step 2 : Both the variables violate their upper bounds and the choice by (1) is bracketed. We arbitrarily choose x_1.

Step 3 : We (i) replace value of x_1 by $u_1 - x_1^o = -3$ (ii) replace variable x_1 by x_1' (iii) change $c_1 = 3$ to $c_1' = -3$ (iv) change sign of all entries in the first row except the sign of "1" in (1,1) position. The tableau 5 after affecting these transformations reduces to tableau 6.

(Tableau 6)

$c_j \rightarrow$		-3	2	0	0
c_B	x_B	x_1'	x_2	x_3	x_4
-3	$x_1' = -3$	1	0	$\boxed{-\frac{5}{29}}$	$\frac{2}{29}$
2	$x_2 = 10$	0	1	$\frac{2}{29}$	$\frac{5}{29}$
	50	0	0	$\frac{19}{29}$	$\frac{4}{29}$

Step 4 : The optimal solution by dual simplex method [76] is given in tableau 7.

(Tableau 7)

$c_j \rightarrow$		-3	2	0	0
c_B	x_B	x_1'	x_2	x_3	x_4
0	$x_3=\frac{87}{5}$	$-\frac{29}{5}$	0	1	$-\frac{2}{5}$
2	$x_2=\frac{44}{5}$	$\frac{2}{5}$	1	0	$\frac{1}{5}$
	$\frac{193}{5}$	$\frac{19}{5}$	0	0	$\frac{2}{5}$

Iteration 1

Step 2 : The variable $x_2^0 = \frac{44}{5}$ exceeds its upper bound 7.

Step 3 : Performing substitution steps we get the tableau 8 below.

(Tableau 8)

$c_j \rightarrow$		-3	-2	0	0
c_B	x_B	x_1'	x_2'	x_3	x_4
0	$x_3=\frac{87}{5}$	$-\frac{29}{5}$	0	1	$-\frac{2}{5}$
-2	$x_2'=-\frac{9}{5}$	$-\frac{2}{5}$	1	0	$\boxed{-\frac{1}{5}}$
	$\frac{193}{5}$	$\frac{19}{5}$	0	0	$\frac{2}{5}$

Step 4 : Dual simplex algorithm gives the optimal solution as in tableau 9.

(Tableau 9)

	$c_j \rightarrow$	-3	-2	0	0
c_B	x_B	x_1'	x_2'	x_3	x_4
0	$x_3=21$	-5	-2	1	0
0	$x_4=9$	2	-5	0	1
	35	3	2	0	0

Iteration 2

Step 2 : All the variables are within their bounds.

Optimal solution is $x_1=7$, $x_2=7$, $z=35$.

CHAPTER - 5

ASSIGNMENT POLYTOPE

5.1 Introduction

The assignment problem of order n is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\text{(AP)} \qquad \sum_{j=1}^{n} x_{ij} = 1 \quad , \quad i=1,2,\ldots,n.$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad , \quad j=1,2,\ldots,n.$$

$$x_{ij} = 0 \text{ or } 1$$

Associated with this (LPP) Linear Programming Problem (without integer restrictions) is the assignment polytope (AP_n) of order n which is the set of all feasible solutions to the assignment problem. Justification for the formulation of assignment problem as LPP lies in the fact that the extreme points of AP_n, thought of as nxn matrices are precisely the permutation matrices (assignments). For proof refer to [92]. This also follows from the concept of total unimodularity of matrices [29,58]. Many methods

are available for solving the assignment problem. See for example [11,49,98,117].

The structure of the assignment polytope helps to understand the assignment problem more clearly. A general study of polytopes is given in [71]. Klee and Witzgall [96] have discussed various aspects of transportation polytopes of which assignment polytope is a special case. However there are many results which are either not true for the general transportation polytopes or are not known.

In this chapter, we study the adjacency of the vertices of the assignment polytope, the structure of its faces and some properties of the assignment polytope. Our approach is completely elementary and simpler than the work of some other research workers in the same direction [12].

5.2 Adjacency Results

Before we derive the adjacency results we need some definitions and notations.

With any permutation $\pi \in S_n$, we associate a permutation matrix $x(\pi) = \{x_{ij}(\pi)\}$ as follows :

$$x_{ij}(\pi) = 1 \quad \text{if } \pi(i) = j$$

$$= 0 \quad \text{otherwise.}$$

Thus, whereever convenient, we may identify permutation matrix $x(\pi)$ with permutation π only. Furthermore, we may use sometimes the notation $i_1 i_2 \ldots i_n$ to mean the permutation $\begin{pmatrix} 1 & 2 & \ldots n \\ i_1 & i_2 & \ldots i_n \end{pmatrix}$ e.g. $\begin{pmatrix} 1234 \\ 2143 \end{pmatrix}$ will either be written (in usual notation) as (12)(34) or as 2143. This notation in which the first row in its natural order is understood and hence not written is sometimes referred as "two rowed notation". Let π and σ be any two permutations. Define

$$I = \left\{(i,j) \mid x_{ij}(\pi) = 0 \text{ and } x_{ij}(\sigma) = 0\right\}$$

Let $c = \{c_{ij}\}$ be an nxn matrix such that

$$c_{ij} = 1 \quad \text{if } (i,j) \in I$$

$$= 0 \quad \text{if } (i,j) \notin I$$

Define the hyperplane H as

$$H : \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} = 0$$

which may shortly be written as $H : cx = 0$ (the manner of multiplication being understood). The line segment joining $x(\pi)$ and $x(\sigma)$ is denoted as $[x(\pi), x(\sigma)]$ or simply as $[\pi, \sigma]$.

<u>Theorem 1</u> : H is a supporting hyperplane to AP_n containing $x(\pi)$ and $x(\sigma)$.

<u>Proof</u> : For any $x = \{x_{ij}\} \in AP_n$

$$cx = \sum_{i,j} c_{ij}\, x_{ij} \geq 0$$

and $cx = 0$ for $x = x(\pi)$, $x(\sigma)$ (by definition of c). This completes the proof.

Using any two permutation π and σ, we can construct a supporting hyperplane as above. This hyperplane will be used throughout this chapter (and the next chapter also). The intersection of this supporting hyperplane (w.r.t. π and σ) with AP_n i.e. $H \cap AP_n$ is referred as the <u>face $AF(\pi,\sigma)$</u> of AP_n. We observe that $AF(\pi,\sigma)$ is convex and π and σ are in $AF(\pi,\sigma)$.

We now prove a theorem which characterises the adjacency of any two permutations on AP_n in terms of their adjacency on $AF(\pi,\sigma)$.

<u>Theorem 2</u> : Two permutations π and σ are adjacent on AP_n iff π and σ are adjacent on $AF(\pi,\sigma)$.

<u>Proof</u> : If π and σ are adjacent on AP_n then $[\pi,\sigma]$ is an edge of AP_n.

$\Rightarrow$ Any point $x \in [\pi,\sigma]$ has a unique representation as a convex combination of the vertices of AP_n and that unique representation is given by

$$x = \alpha x(\pi) + (1-\alpha)x(\sigma) \text{ for some } \alpha \text{ such that}$$

$0 \leq \alpha \leq 1$.

$\Rightarrow$ $x \in [\pi, \sigma]$ has unique representation as convex combination of the vertices of $AF(\pi, \sigma)$.

$\Rightarrow$ $[\pi, \sigma]$ is an edge of $AF(\pi, \sigma)$.

$\Rightarrow$ $[\pi, \sigma]$ are adjacent on $AF(\pi, \sigma)$.

<u>Conversly</u> Let $AF(\pi, \sigma)$ have m vertices, namely, $T_1, T_2, \ldots, T_m$ such that $T_1 \equiv \pi$, $T_2 \equiv \sigma$ and π and σ are adjacent on $AF(\pi, \sigma)$. This implies any point x on the line segment $[\pi, \sigma]$ has a unique representation as a convex combination of the vertices of $AF(\pi, \sigma)$ and that unique representation is given by

$$x = \alpha_1 x(\pi) + \alpha_2 x(\sigma) \text{ for some } \alpha_1, \alpha_2 \text{ such that } \alpha_1 \geq 0, \alpha_2 \geq 0$$
$$\alpha_1 + \alpha_2 = 1.$$

$$\text{Let } x = \sum_{i=1}^{n!} \alpha_i x(T_i) \quad , \alpha_i \geq 0, \quad \sum_{i=1}^{n!} \alpha_i = 1 \quad \ldots(1)$$

be any representation of x as convex combination of vertices of AP_n where vertices of AP_n are named as $T_1, T_2, \ldots, T_{n!}$ such that $T_1 \equiv \pi$, $T_2 \equiv \sigma$. Since x lies on the hyperplane H, therefore

$$0 = cx = \sum_{i=1}^{n!} \alpha_i cx(T_i) = \sum_{i=1}^{m} \alpha_i \; cx(T_i) + \sum_{i=m+1}^{n!} \alpha_i \; cx(T_i) \quad \ldots(2)$$

Since H is a supporting hyperplane to AP_n and $T_i \notin AF(\pi, \sigma)$ for $i > m$,

$$cx(T_i) > 0 \text{ for } i = m+1, \ldots, n.$$

Thus in order that (2) may hold $\alpha_i = 0$, $i = m+1,\ldots,n$ and this reduces (1) to the following form

$$x = \sum_{i=1}^{m} \alpha_i \, x(T_i), \quad \alpha_i \geq 0, \quad \sum_{i=1}^{m} \alpha_i = 1$$

Now, since π and σ are adjacent on $AF(\pi, \sigma)$, therefore

$$x = \alpha_1 \, x(\pi) + \alpha_2 x(\sigma) \quad \text{(recalling that } \pi \equiv T_1, \ \sigma \equiv T_2)$$

i.e. x in (1) is expressible as a convex combination of $x(\pi)$ and $x(\sigma)$ only which implies π, σ are adjacent on AP_n.

The importance of the above theorem lies in the fact that adjacency of any two vertices on AP_n can be decided by deciding their adjacency on $AF(\pi, \sigma)$ which consists of fewer vertices of AP_n. The following theorem relates the nature of π and σ to the number of vertices of $AF(\pi, \sigma)$.

<u>Theorem 3</u> : If π and σ are such that $\sigma\pi^{-1}$ is a product of p disjoint cycles of length > 1 then $AF(\pi, \sigma)$ contains 2^p permutations (including π and σ).

<u>Proof</u> : Let the p disjoint cycles of $\sigma\pi^{-1}$ be

$$S_1, S_2, \ldots, S_p$$

such that each $S_k, k = 1,2,\ldots,p$ has length greater than one.

Let $S = \bigcup_{k \in Q} S_k$. where $Q \subseteq \{1,2,\ldots,p\}$

Define $J = \{\pi^{-1}(i) \mid i \in S\}$

Construct a permutation T with respect to set S as follows :

$$T(j) = \sigma(j), \quad \text{if } j \in J$$

$$T(j) = \pi(j), \quad \text{if } j \notin J$$

We show that $T \in AF(\pi, \sigma)$.

$I = \{1,2,\ldots,n\}$ and let $K = \{j \in I \mid \sigma(j) = \pi(j)\}$

i.e. set of indices corresponding to cycles of length one. Then $T(j) = \sigma(j) = \pi(j)$ for $j \in K$.

Also for $j \in J$, we have $T(j) = \sigma(j)$ and for $j \in I - (J \cup K)$, we have $T(j) = \pi(j)$

$$\sum_i \sum_j c_{ij}\, x_{ij}(T) = \sum_{i \in K} \sum_j c_{ij}\, x_{ij}(T) + \sum_{i \in J} \sum_j c_{ij}\, x_{ij}(T) + \sum_{i \in I-(J \cup K)} \sum_j c_{ij}\, x_{ij}(T)$$

and each of the three summations on right hand side is zero separately.

Therefore $cx(T) = 0$ which implies $T \in AF(\pi, \sigma)$.

For different choices of set S we can construct different permutations T belonging to $AF(\pi, \sigma)$. Since there are 2^p choices of Q and each choice of Q gives a permutation T, we get 2^p permutations in $AF(\pi, \sigma)$.

To prove that these are the only permutations $\in AF(\pi, \sigma)$, let T be any permutation such that $cx(T) = 0$. Clearly $cx(T) = 0$ should hold separately for each row j of matrix c and x(T) (by definition of c)

$\Rightarrow T(j) = \sigma(j) = \pi(j)$ for $j \in K$.
and $T(j) = \sigma(j)$ or $\pi(j)$ for $j \notin K$.

Let $J = \{j | T(j) = \sigma(j)\}$

Thus for $j \in I - (J \cup K)$, $T(j) = \pi(j)$

Let $S = \{\sigma(j) | j \in J\}$

If $S_1 = \{\pi(j) | j \in J\}$

Then $S = S_1$, because if not, then there exists at least one $j \in J$, say, $j = t_1$, such that $\sigma(t_1) \in S$ and $\sigma(t_1) \notin S_1$

$\Rightarrow T(t_1) \in S$ and $T(t_1) \notin S_1$ (because $T(t_1) = \sigma(t_1)$)

$\Rightarrow T(t_1) \neq \pi(j)$ for $j \in J$

$\Rightarrow T(t_1) = \pi(j)$ for some $j \in I - (J \cup K)$

$\Rightarrow T(t_1) = T(j)$ for some $j \in I - (J \cup K)$ (because $T(j) = \pi(j)$ for $j \in I - (J \cup K)$)

$\Rightarrow$ T is not a permutation. Hence $S = S_1$.
Also $\sigma(j) \neq \pi(j)$ for $j \in J$. Therefore S is a union of

disjoint cycles of $\sigma\pi^{-1}$ such that each cycle is of length greater than one i.e. $S = \bigcup_{k \in Q} S_k$. Therefore any $T \in AF(\pi, \sigma)$ is obtainable from some set S of the above type only which implies $AF(\pi, \sigma)$ has precisely 2^p permutations and they can be constructed by the knowledge of the cycles of $\sigma\pi^{-1}$.

Remark 1 : It can be observed that 2^p permutations of $AF(\pi, \sigma)$ can also be generated by premultiplying π by cycles of $\sigma\pi^{-1}$, the combinations of cycles being taken in all possible ways. Thus members of $AF(\pi, \sigma)$ are

$S_1\pi,\ S_2\pi, \ldots, S_p\pi,\ S_1S_2\pi, S_1S_3\pi, \ldots, S_1S_p\pi,\ S_2S_3\pi, \ldots, S_2S_p\pi, \ldots$
...., and these are precisely 2^p in number.

Corollary 1 : An immediate consequence of the above Theorem is that if π and σ be two optimal assignments such that $\sigma\pi^{-1}$ is product of p disjoint cycles of length greater than one, then there are at least 2^p optimal assignments.

The following Theorems (4,5 and 6) decide the adjacency of two permutations and the number of vertices (order of adjacency) adjacent to a given vertex on AP_n.

<u>Theorem 4</u> : If π and σ be any two permutations such that $\sigma\pi^{-1}$ is product of p disjoint cycles then 2^p permutations $\in AF(\pi,\sigma)$ can be so paired such that 2^{p-1} line segments obtained by joining these pairs have the same mid point as the line segment $[x(\pi),x(\sigma)]$.

<u>Proof</u> : Let $T_1 \in AF(\pi,\sigma)$ be obtained through $Q \subseteq \{1,2,\ldots,p\}$ as in Theorem 3. Let $T_2 \in AF(\pi,\sigma)$ be similarly obtained through $Q' = \{1,2,\ldots,p\} - Q$. Then obviously ;

$$\frac{1}{2}\,x(T_1) + \frac{1}{2}\,x(T_2) = \frac{1}{2}\,x(\pi) + \frac{1}{2}\,x(\sigma)$$

This completes the proof.

Each of the 2^{p-1} pairs of assignments obtained in Theorem 4 are referred to as <u>complimentary pair of assignments</u> (or simply <u>complimentary assignments</u>) w.r.t. the given pair of assignments π and σ.

<u>Theorem 5</u> : Two permutation π and σ are adjacent on AP_n iff $\sigma\pi^{-1}$ is cyclic.

<u>Proof</u> : We know by Theorem 2 that two permutations π and σ are adjacent on AP_n iff they are adjacent on $AF(\pi,\sigma)$. This is so iff $AF(\pi,\sigma)$ contains just two permutations (π and σ itself) which happens iff $\sigma\pi^{-1}$ is cyclic because otherwise π and σ are non-adjacent in view of Theorem 4.

<u>Theorem 6</u> : The number of extreme points adjacent to any given extreme point on AP_n is the same and equals

$$N(n) = \sum_{r=2}^{n} (r-1)!\, {}^{n}c_{r}$$

<u>Proof</u> : That the order of adjacency for each vertex is the same is obvious. To compute this number, let $\pi = 1,2,\ldots,r,\ldots,n$ be any permutation in two rowed notation. If σ is adjacent to π then σ must be of the form so as to form a cycle of length r with π (σ written in two rowed notation). There are ${}^{n}c_{r}$ choices of cycles of length r. Keeping one element fixed, any such cycle can be permutated among its elements in $(r-1)!$ ways to give rise to different permutations. Also since minimum length of a cycle to give rise to $\sigma \neq \pi$ is 2, therefore

$$N(n) = \sum_{r=2}^{n} (r-1)!\, {}^{n}c_{r}$$

Here below we give some numerical estimates of the adjacency which, as clear from the tableau, is of very high order even for the assignment polytopes of small order. The table also gives the dimension of AP_n. We know that nxn assignment problem when written as linear program has 2n rows out of which any 2n-1 rows are linearly independent [76, Chapter 9, Transportation Problems] and since AP_n lies in R^{n^2} space, AP_n is linear

vector space of dimension $n^2-(2n-1) = (n-1)^2$. The following table gives the value of dimension of AP_n and $N(n)$ for $1 \leq n \leq 10$.

n	Dimension	No of ext.pts.	No. of adjacent ext.pts.
1	0	1	0
2	1	2	1
3	4	6	5
4	9	24	20
5	16	120	84
6	25	720	409
7	36	5040	2540
8	49	40320	16064
9	64	362880	125664
10	81	3628800	1112073

(Some Estimates for AP_n)

In the following Theorems, 7 and 8, we mention some more facts about the face $AF(\pi, \sigma)$ which helps us to understand the structure of AP_n more clearly.

<u>Theorem 7</u> : If each cycle of $\sigma\pi^{-1}$ is of length, k, then the distance between any two extreme points adjacent over $AF(\pi, \sigma)$ is $(2k)^{1/2}$.

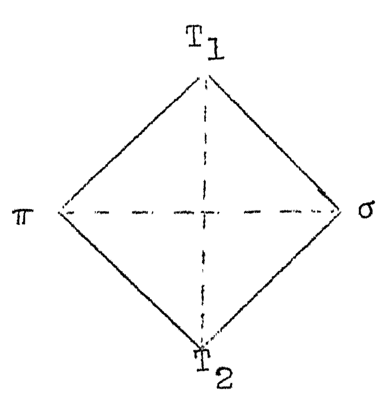
Proof : If T_1 and T_2 be any two adjacent permutations over $AF(\pi, \sigma)$, then $T_1 T_2^{-1}$ is cyclic. Since T_1 and T_2 are obtainable from the knowledge of cycles of $\sigma\pi^{-1}$ therefore $T_1 T_2^{-1}$ is some cycle of $\sigma\pi^{-1}$ whose length is given to be k.

$$\text{Obviously } \|x(T_1)-x(T_2)\| = (2k)^{\frac{1}{2}}$$

Theorem 8 : If $\sigma\pi^{-1}$ is product of p disjoint cycles then any permutation $T_1 \in AF(\pi, \sigma)$ has exactly p permutations $\in AF(\pi, \sigma)$ which are adjacent over $AF(\pi, \sigma)$.

Proof : Let T_1 and T_2 be two permutations over $AF(\pi, \sigma)$. They are adjacent over $AF(\pi, \sigma)$ iff $T_1 T_2^{-1}$ is cyclic. Also cycle of $T_1 T_2^{-1}$ is some cycle of $\sigma\pi^{-1}$. Obviously such choices of T_2 in $AF(\pi, \sigma)$ for a fixed T_1 are precisely p. Hence the proof.

Geometrically we can think of face $AF(\pi, \sigma)$ as in Figure 1 below. The figure also demonstrates the complimentary character of T_1 and T_2 such that mid point of $[x(T_1), x(T_2)]$ being the same as the mid point of $[x(\pi), x(\sigma)]$.



(Figure 1)

However all the faces of AP_n are certainly not of the kind considered above. For example, there can be faces of AP_n with only 3 vertices on them as the following example shows.

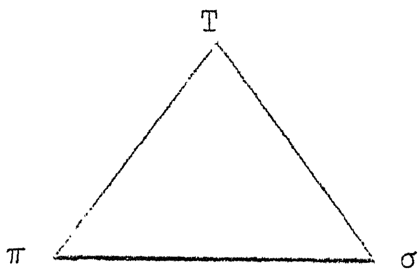
Example

$$\pi = 123456$$

$$\sigma = 134526 \quad \text{(Two rowed notation)}$$

$$T = 134256$$

We observe $\pi\sigma^{-1} = (2345)$, $\sigma T^{-1} = (52)$, $\pi T^{-1} = (234)$ so that π, σ and T are adjacent to one another over AP_n giving rise to a triangular face of AP_n (Figure 2)



(Figure 2)

We now give below few properties of the inverses of the permutations and the relation between their corresponding faces.

Clearly a permutation π and its inverse π^{-1} are adjacent.

<u>Theorem 9</u> : If π and σ be two permutations adjacent over AP_n then so are π^{-1} and σ^{-1}. Also distance between π^{-1} and σ^{-1} is the same as distance between π and σ .

<u>Proof</u> : Let $\pi : \pi(1),\pi(2),\ldots,\pi(k),\ldots,\pi(n)$

$\sigma : \sigma(1),\sigma(2),\ldots,\sigma(k),\ldots,\sigma(n)$

Let $I = \{i : \pi(i) \neq \sigma(i)\}$

Without loss of generality, let $I = \{1,2,\ldots,k\}$ so that $\pi(i) = \sigma(i)$ for $i = k+1,\ldots,n$. Again, we assume without loss of generality

$$\sigma(i) = \pi(i+1) \quad , \; i = 1,\ldots,k-1$$
$$\sigma(k) = \pi(1)$$

For $i \notin I$, $\pi^{-1}\sigma(i) = \pi^{-1}\pi(i) = i$

For $i \in I$, $\pi^{-1}\sigma(i) = \pi^{-1}\pi(i+1)$
$= i+1 \qquad (i = 1,\ldots,k-1)$

and $\pi^{-1}\sigma(k) = 1$

$\Longrightarrow$ $\pi^{-1}\sigma$ is cyclic of length k. Thus π^{-1} and σ^{-1} are adjacent over AP_n.

Obviously $||x(\pi^{-1})-x(\sigma^{-1})|| = ||x(\pi)-x(\sigma)||$

<u>Corollary 2</u> : Permutations π and σ are adjacent over AP_n iff π^{-1} and σ^{-1} are adjacent over AP_n.

<u>Proof</u> : If π and σ are adjacent, then each of $\sigma\pi^{-1}, \pi\sigma^{-1}, \pi^{-1}\sigma, \sigma^{-1}\pi$ is cyclic. If π and σ are non-adjacent then π^{-1} and σ^{-1} are also non-adjacent and accordingly each of $\sigma\pi^{-1}, \pi\sigma^{-1}, \pi^{-1}\sigma$, $\sigma^{-1}\pi$ is non-cyclic and is product of same number of disjoint cycles.

The proof now follows obviously.

The following theorem establishes the relation between the faces corresponding to inverse permutations.

<u>Theorem 10</u> : If π and σ be two permutations over AP_n and $AF(\pi, \sigma)$ and $AF(\pi^{-1}, \sigma^{-1})$ be faces of AP_n corresponding to permutation pairs (π, σ) and (π^{-1}, σ^{-1}) respectively, then

$$\text{(a)} \qquad |AF(\pi, \sigma)| = |AF(\pi^{-1}, \sigma^{-1})|$$

$$\text{(b)} \qquad [AF(\pi, \sigma)]^{-1} = AF(\pi^{-1}, \sigma^{-1})$$

where $[AF(\pi, \sigma)]^{-1}$ denotes the set of inverse of permutations in $AF(\pi^{-1}, \sigma^{-1})$.

<u>Proof</u> : Part (a) is obvious because if $\sigma \pi^{-1}$ is product of p disjoint cycles then so is $\sigma^{-1}\pi$.

If π and σ are two adjacent permutations then part (b) also follows obviously. So we assume that π and σ are two non-adjacent permutations so as $\sigma \pi^{-1}$ is product of, say, p disjoint cycles of length greater than one. Let $T_1 \in AF(\pi, \sigma)$ be obtainable through S such that

$$S = \bigcup_{k \in Q} S_k \;, \qquad Q \subseteq \{1, 2, \ldots, p\}$$

$$J = \{\pi^{-1}(i) \mid i \in S\}$$

Let $S' = I-S$ and $J^* = \{\pi^{-1}(i) | i \in S'\}$

Obviously $J \cap J^* = \varphi$ and $J \cup J^* = I$

Thus $T_1(j) = \sigma(j)$, $j \in J$

$T_1(j) = \pi(j)$, $j \in J^*$

Let $J_1 = \{j \in J | \pi^{-1}(j) \in J\}$ and $J_2 = \{j \in J | \pi^{-1}(j) \notin J\}$

We observe $J_1 \cap J_2 = \varphi$ and $J_1 \cup J_2 = J$

Let $J_3 = \{j \in J^* | \pi^{-1}(j) \in J^*\}$ and $J_4 = \{j \in J^* | \pi^{-1}(j) \notin J^*\}$

We observe $J_3 \cap J_4 = \varphi$ and $J_3 \cup J_4 = J^*$

Now define $T_1^* \in AF(\pi^{-1}, \sigma^{-1})$ as follows :

$$T_1^*(j) = \sigma^{-1}(j) \quad , j \in J_1 \cup J_4$$

$$T_1^*(j) = \pi^{-1}(j) \quad , j \in J_2 \cup J_3$$

We prove in the following that $T_1^* = T_1^{-1}$

For $j \in J_1, T_1T_1^*(j) = T_1(\sigma^{-1}(j))$ (for $j \in J_1, \pi^{-1}(j) \in J_1$

$= \sigma(\sigma^{-1}(j))$ $\Longrightarrow \sigma^{-1}(j) \in J_1$)

$= j$

For $j \in J_2, T_1T_1^*(j) = T_1(\pi^{-1}(j))$ (for $j \in J_2, \pi^{-1}(j) \notin J$

$= \pi(\pi^{-1}(j))$ $\Longrightarrow \pi^{-1}(j) \in J^*$)

$= j$

For $j \in J_3, T_1T_1^*(j) = T_1(\pi^{-1}(j))$ (for $j \in J_3, \pi^{-1}(j) \in J^*$)

$= \pi(\pi^{-1}(j))$

$= j$

For $j \in J_4, TT_1^*(j) = T(\sigma^{-1}(j))$ (for $j \in J_4, \pi^{-1}(j) \notin J^*$

$= \sigma(\sigma^{-1}(j))$ $= \sigma^{-1}(j) \notin J^*$

$= j$ $= \sigma^{-1}(j) \in J$)

which proves $T_1^* = T_1^{-1}$.

Since $|AF(\pi^{-1}, \sigma^{-1})| = |AF(\pi, \sigma)|$, (by part a)
the proof of part (b) is complete.

5.3 Diameter of Assignment Polytope

The distance d(a,b) between a pair of vertices a and b of a convex polytope is the minimum number of steps needed to reach vertex b starting from vertex a and always moving along an edge of the polytope. (A step means moving from one extreme point to another adjacent extreme point along an edge of the polytope). The diameter d of a polytope X is defined as the greatest distance between any pair of vertices of the polytope X i.e.

$$d = \max_{a,b \in X} d(a,b)$$

Theorem 11 : The diameter of AP_n is two.

Proof : To prove that the diameter of AP_n is two, we prove that if π and σ be two non-adjacent vertices of AP_n it is possible to construct a vertex T of AP_n adjacent to both π and σ.

Since π and σ are non-adjacent, $\sigma\pi^{-1}$ is product of, say, p disjoint cycles $S_1, S_2, \ldots, S_p$.

If we constract a permutation T as in Theorem 3 by taking S to be one of the cycles $S_1, S_2, \ldots, S_p$, then

clearly πT^{-1} and σT^{-1} are cyclic which implies T is adjacent to both π and σ. In general, we can get 2p such permutations (for $p \geqslant 3$) adjacent to both π and σ. However, this number is not enhaustive.

In [12,126], the diameter of AP_n is shown to be equal to two. In [121], the diameter of the assignment polytope, traveling salesman polytope and polytopes associated with a large number of combinatorial problems has been shown to be less than or equal to two. In the case of assignment polytope we have achieved the similar result above in a straightforward manner.

5.4 Graph of $G(AP_n)$

Before we deal with the main result of this section, we give some definitions [78] of graph theory.

The paths in a Graph G are called edge-disjoint paths if they do not have any edge in common. A graph is edge-N-connected provided there exist N mutually edge-disjoint paths between every pair of nodes of the graph G. The distance d(a,b) between a pair of nodes a and b is the length of shortest path containing them. The diameter D(G) of a connected graph G is the greatest distance between any pair of nodes i.e.

$$D(G) = \max_{a,b \in G} d(a,b)$$

Balinski's Conjecture [12]

If $G(AP_n)$ is the graph whose nodes are the vertices of AP_n and edges are the lines join-ing the two adjacent vertices of AP_n, then $G(AP_n)$ is $N(n)$-connected where $N(n)$ given by

$$N(n) = \sum_{r=2}^{n} {}^{n}C_{r} \, (r-1)!$$

is the number of vertices adjacent to any vertex x on AP_n.

Weaker Form of the Conjecture* [89]

$G(AP_n)$ is edge-$N(n)$-connected.

We will first prove this weaker form of the conjecture and then a small observation will extend the result to all polytopes of diameter two, a large class of which is considered in [121].

Proof : For any vertex a of AP_n, let

$$X(a) = \{x \in AP_n : x \text{ is adjacent to } a\}$$

For any two non adjacent vertices a and b of AP_n let

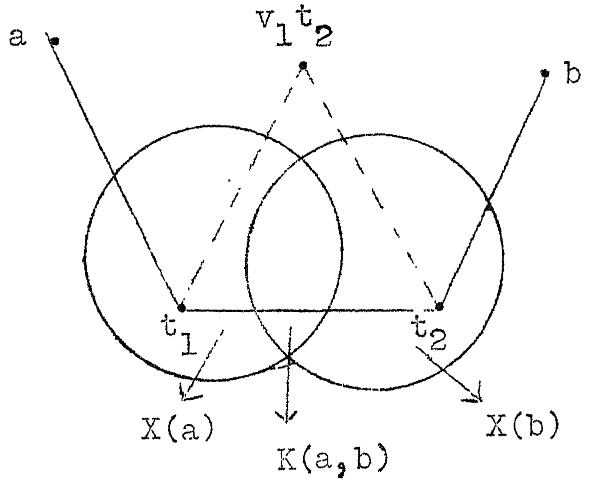
$$X(a) \cap X(b) = K(a,b)$$

*Presented at Ninth Annual Convention of Operation Research Society of India held at Calcutta(India) in Dec.1976 and was awarded the prize for the best student paper [89]..

$$K_1(a,b) = X(a)-K(a,b)$$

$$K_2(a,b) = X(b)-K(a,b)$$

There are three types of paths as described below :



(Figure 3)

(I) For $k \in K(a,b)$

$$P(a,k,b) = [(a,k),(k,b)] \qquad \ldots(3)$$

is a path which joins a to b.

For $k_1 \in K(a,b)$ and $k_2 \in K(a,b)$ the paths $P(a,k_1,b)$ and $P(a,k_2,b)$ are two disjoint paths joining a to b.

Thus

$$S_1 = [\, P(a,k,b) \; : \; k \in K(a,b) \,]$$

is a set of disjoint paths joining a to b.

(II) Let $S = [(t_1,t_2) \; : \; t_1 \in K_1(a,b),\ t_2 \in K_2(a,b),$
t_1 is adjacent to $t_2]$.

Let $S^* \subseteq S$ such that

$$(t_1,t_2) \in S^* \text{ and } (t_1,t_2') \in S^* \implies t_2 = t_2'$$

$$(t_1,t_2) \in S^* \text{ and } (t_1',t_2) \in S^* \implies t_1 = t_1'$$

For $(t_1,t_2) \in S^*$

$$P(a,t_1,t_2,b) = [(a,t_1),(t_1,t_2),(t_2,b)] \qquad \ldots(4)$$

is a path which joins a to b.

For $(t_1,t_2) \in S^*$, $(t_3,t_4) \in S^*$, the paths $P(a,t_1,t_2,b)$ and $P(a,t_3,t_4,b)$ are disjoint paths joining a to b.

Thus

$$S_2 = [\, P(a,t_1,t_2,b) \; : \; (t_1,t_2) \in S^* \,]$$

is a set of disjoint paths joining a to b. Observe that each path of S_2 is also disjoint from each path of S_1.

Define

$$S_1^* = [\, t_1 \; : \; (t_1,t_2) \in S^* \text{ for some } t_2 \in K_2(a,b) \,]$$

$$S_2^* = [\, t_2 \; : \; (t_1,t_2) \in S^* \text{ for some } t_1 \in K_1(a,b) \,]$$

Let

$$K = K_1(a,b)-S_1^*$$

$$P = K_2(a,b)-S_2^*$$

Obviously, K and P have the same number of elements and for any $t_1 \in K$, there exist no $t_2 \in P$ such that

t_1 and t_2 are adjacent. ...(5)

III Now for $T_1' \in K$, chooses any $t_2 \in P$

Since t_1, t_2 are non adjacent

$$t_1 t_2^{-1} \text{ is not cyclic.}$$

Now every permutation which is not cyclic can be written as a product of two cucles [12] i.e. if π is product of k disjoint cycles

$$\pi = (\pi(1),\ldots,\pi(n_1))\ (\pi(n_1+1),\ldots,\pi(n_2))\ldots(\pi(n_{k-1}+1)\ldots\pi(n))$$

then

$$\pi = (\pi(1)\ldots\pi(n))(\pi(n)\ldots\pi(n_2)\ \pi(n_1))$$

$$= uv$$

where $u = (\pi(1)\ldots\pi(n))$ and $v = (\pi(n)\ldots\pi(n_2)\pi(n_1))$ are both cyclic.

Let

$$t_1 t_2^{-1} = u_1 v_1$$

where u_1 and v_1 are cyclic

Obviously $v_1 t_2$ is adjacent to both t_1 and t_2.

Then

$$P(a,t_1,v_1t_2,t_2,b) = [(a,t_1),(t_1,v_1t_2), (v_1t_2,t_2),(t_2,b)] \quad \ldots.(6)$$

is a path joining a to b .

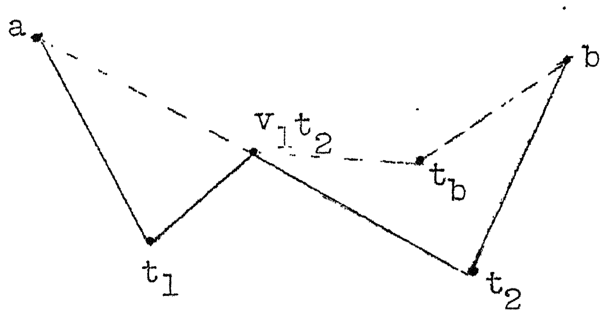
We observe that

$$v_1t_2 \notin K \cup P$$

(a) If $v_1t_2 \in (K_1(a,b) \cup K_2(a,b))'$, then we are through

(b) If $v_1t_2 \in (K_1(a,b)-K)$, then path (6) is disjoint from path of the type (4), namely,

$$(a,v_1t_2), (v_1t_2,t_b), (t_b,b)$$



(Figure 4)

where t_b is adjacent to $v_1t_2, t_b \in (K_2(a,b)-P)$ and $t_b \neq t_2$ otheriwse $(v_1t_2,t_2) \in S^*$ and thus $t_2 \in S_2^*$

$$\Rightarrow \quad t_2 \notin P$$

and this contradicts the choice of t_2 made in (5).

(c) If $v_1t_2 \in (K_2(a,b)-P)$, the argument is similar as in case of (b) and (6) is a path disjoint to the paths of the type (4).

For $t_3 \in K$, $t_4 \in P$, the paths

$$P(a,t_1,v_1t_2,t_2,b) \text{ and } P(a,t_3,v_1t_4,t_4,b)$$

are two disjoint paths joining a to b.

Thus set $S_3 = [P(a,t_1,v_1t_2,t_2,b) : t_1 \in K,\ t_2 \in P]$ is a set of disjoint paths joining a to b. Observe that each path in S_3 is disjoint from each path in S_1 and S_2 and the number of elements in $S_1 \cup S_2 \cup S_3$ equals $N(n)$.

If a and b are adjacent vertices than (a,b) constitutes one path and the procedure outlined above for non-adjacent case can be repeated by replacing $X(a)$ and $X(b)$ by $X(a)-\{b\}$ and $X(b)-\{a\}$ respectively.

Examples

For the purpose of demonstration, the following simple examples of the assignment polytope are considered.

For $n \leq 3$ we have paths of length one only. For $n = 4$, it can be easily seen that any two non-adjacent vertices π and σ on P_4 are such that

$$X(\pi) = X(\sigma) = K(\pi,\sigma)$$

Remark 2 : In the above proof, the number of vertices adjacent to any vertex on the assignment polytope AP_n is the same. However, this may not be true in general. In case of any polytope P.

$$\text{Define} \qquad N = \min_{a \in P} N(a)$$

Then G(P) is edge-N-connected. The proof will be exactly similar.

CHAPTER - 6

TRAVELING SALESMAN POLYTOPE

6.1 Introduction

The traveling salesman problem of order n is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

(TSP) subject to

$$\sum_{j=1}^{n} x_{ij} = 1 , \quad i = 1,2,..,n$$

$$\sum_{i=1}^{n} x_{ij} = 1 , \quad j = 1,2,..,n$$

$$x_{ij} = 0 \quad \text{or } 1$$

and the solution set of those x_{ij}'s for which $x_{ij} = 1$ forms a tour where a tour is defined as the cyclic permutation $(i_1, i_2, \ldots, i_n)$ such that the solution $x_{i_1 i_2} = x_{i_2 i_3} = \ldots = x_{i_n i_1} = 1$ and zeroes elsewhere and vice versa. Evidently there are $(n-1)!$ tours which are all feasible solutions of (TSP).

Associated with this traveling salesman problem of order n is the traveling salesman polytope (TP_n) of order n which is defined as the convex hull of permutation

matrices corresponding to traveling salesman tours (or tours simply). Evidently TP_n is a collapsed polytope of AP_n. Murty [118] has discussed the adjacency properties of extreme points of collapsed polytopes. If two extreme points are adjacent over a polytope, they are also adjacent over its collapsed polytope. However the converse is not true. Thus if π and σ are adjacent over AP_n, then π and σ are adjacent over TP_n.

Many exact as well as heuristic methods for traveling salesman problem have been developed. See for example [39,49,58,79,80]. However, in all the existing methods, the computational time increases exponentially with increase in the size of problem. In fact it has been shown that the traveling salesman problem is NP-hard. Refer to Karp [93] for details. However, Rosenkrantz, Stearns and Lewis II [132] have considered some polynomial time algorithms for finding " ϵ approximate optimal tours", but not necessary optimal tours, for the traveling salesman problem.

One good reason that good methods could not be available can be that very little is known to-day about the traveling salesman polytope. In this chapter, we make an attempt to study the traveling salesman

polytope. Primarily, we concern ourselves with the adjacency of the tours on TP_n. The approach developed for AP_n in chapter 5, is further explored over TP_n to know the adjacency of the tours on TP_n. The study, however, is incomplete.

In section 6.2, we derive some adjacency results over TP_n. Section 6.3 concerns with some adjacency rules and their applications for the development of heuristics. Section 6.4 lists some problems for future research.

6.2 Adjacency Results

Corresponding to tours π, σ we define the face $TF(\pi, \sigma)$ of TP_n, similar to face $AF(\pi, \sigma)$ of AP_n as follows :

$$TF(\pi, \sigma) = \left\{ T : cx(T) = 0 \text{ and } T \text{ is a tour} \right\}$$

where the matrix c is constructed w.r.t. tour π, σ as in chapter 5.

For a tour π, let tour σ be such that $\sigma\pi^{-1}$ is product of p cycles each of length > 1. Let $S = S_1 \cup S_2 \ldots \cup S_p$ and let $|S| = k$, then we say $\sigma \in Z_k(\pi)$ or simply $\sigma \in Z_k$. Note that $Z_2 = \varphi$.

In view of the following Theorem (similar to Theorem 2 of chapter 5) the adjacency of any two vertices of TP_n can be decided by deciding their adjacency over

$TF(\pi, \sigma)$ which contains much smaller number of vertices than TP_n.

Theorem 1 : Two tours π and σ are adjacent over TP_n iff they are adjacent over $TF(\pi, \sigma)$.

Proof : Exactly similar to Theorem 2 of chapter 5.

Theorem 2 : If $|TF(\pi, \sigma)| \leq 3$, then π and σ are adjacent over TP_n.

Proof : If $TF(\pi, \sigma) = 2$ i.e. $TF(\pi, \sigma) = \{\pi, \sigma\}$ then obviously $[\pi, \sigma]$ is an edge of TP_n.

Let $TF(\pi, \sigma) = \{T, \pi, \sigma\}$. Since T, π and σ are three vertices of TP_n (lying over a hyperplane $H : cx = 0$) they cannot lie over a line. Therefore $[\pi, \sigma]$ is an edge of $TF(\pi, \sigma)$ which implies it is an edge of TP_n. Hence the proof.

In fact we have also proved above that $[T, \pi]$ are $[T, \sigma]$ is also an edge of $TF(\pi, \sigma)$ so that all the three tours of $\{T, \pi, \sigma\}$ are adjacent to one another.

It may be remarked that the above theorem indicates the existence of triangular faces of TP_n. The following example shows the existence of such triangular faces on TP_n.

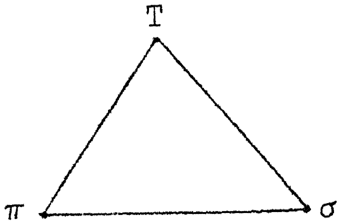
Example 1

$$\text{Let } \pi = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10)$$

$$\sigma = (1\ 7\ 4\ 9\ 3\ 6\ 2\ 5\ 8\ 10)$$

Then $TF(\pi, \sigma) = \{\pi, \sigma, T\}$

where $T = (1\ 2\ 3\ 6\ 7\ 4\ 5\ 8\ 9\ 10)$



(Figure 1)

<u>Theorem 3</u> : If for tours π and σ, $\sigma\pi^{-1}$ is product of two cycles such that at least one of them is of length two then π and σ are adjacent over TP_n.

<u>Proof</u> : Since $\sigma\pi^{-1}$ is product of 2 cycles,

$$|AF(\pi, \sigma)| = 4$$

while π, σ are two permutations in $AF(\pi, \sigma)$, the other two can be obtained through set S_1 <u>only</u> which consists of elements of cycle of length 2. (because a permutation obtained through S can also be obtained through S' by defining $T_1(j) = \pi(j)$, $j \in J'$ and $T_1(j) = \sigma(j)$, $j \notin J'$ where $J' = \{\pi^{-1}(i) | i \in S'\}$. Since $Z_2 = \varphi$, these two permutations do not lead to a tour. Therefore $TF(\pi, \sigma)$ contains π and σ only which implies π, σ are adjacent on TP_n.

<u>Corollary 1</u> : All tours on TP_5 are adjacent to one another.

<u>Proof</u> : If π and σ are permutations on 5 symbols then $\sigma\pi^{-1}$, if not cyclic, can have either two cycles of length two or one cycle of length three and one cycle of length two. In each case, the proof follows by the above Theorem.

The Theorem 2 can be interpreted as saying that a <u>necessary condition</u> for π, σ to be non-adjacent and TP_n is that $|TF(\pi, \sigma)| \geq 4$.

However the above condition is not sufficient as illustrated by the following counter example.

<u>Example 2</u> : Let $\pi = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$, $\sigma = (1\ 4\ 7\ 6\ 3\ 9\ 8\ 2\ 5)$. Using two rowed notation, the elements of the face $TF(\pi, \sigma)$ are written as below :

$$\begin{array}{rl}
\underline{\text{base row}:} & \underline{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9} \\
\pi : & 2\ 3\ 4\ 5\ \underline{6}\ 7\ \underline{8}\ 9\ \underline{1} \\
\sigma : & 4\ 5\ 9\ 7\ \underline{1}\ 3\ \underline{6}\ 2\ \underline{8}
\end{array}$$

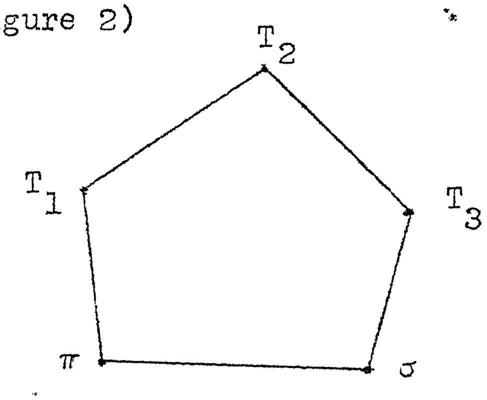
$$\sigma\pi^{-1} = (2\ 4\ 9)\ (1\ 8\ 6)\ (3\ 5\ 7)$$

<u>Elements of $TF(\pi, \sigma)$</u> (other than π and σ) are :

$$\begin{array}{lll}
T_1 : 4\ 3\ 9\ 5\ \underline{6}\ 7\ \underline{8}\ 2\ \underline{1} & = & (1\ 4\ 5\ 6\ 7\ 8\ 2\ 3\ 9) \\
T_2 : 2\ 5\ 4\ 7\ \underline{6}\ 3\ \underline{8}\ 9\ \underline{1} & = & (1\ 2\ 5\ 6\ 3\ 4\ 7\ 8\ 9) \\
T_3 : 4\ 5\ 9\ 7\ \underline{6}\ 3\ \underline{8}\ 2\ \underline{1} & = & (1\ 4\ 7\ 8\ 2\ 5\ 6\ 3\ 9)
\end{array}$$

Now by observing π, σ, T_1, T_2, T_3 and their permutation matrices it is oberved (because of underlined positions i.e. any one of the rows 5th, 7th or 9th of the corresponding permutation matrices) that line segment $[x(\pi), x(\sigma)]$ has a unique representation as a convex combination of extreme points of TP_n. Thus even though $|TF(\pi, \sigma)| \neq 5$, π and σ are adjacent.

Geometrically the face $TF(\pi, \sigma)$ can be viewed as follows (Figure 2)



(Figure 2)

<u>Complementary tours</u> : Two tours T_1 and T_2 on face $TF(\pi, \sigma)$ are called "complementary tours" w.r.t. π and σ or simply "complementary" if they can be obtained through sets S and S' such that $S = \bigcup_{k \in Q} S_k$, $S' = \bigcup_{k \in Q'} S_k$ where $Q \subseteq \{1, 2, \ldots, p\}$ and p is the number of disjoint

cycles of $\sigma\pi^{-1}$ of length greater than one. Thus if T_1 and T_2 are complementary w.r.t. π and σ then

$$\frac{1}{2}x(\pi) + \frac{1}{2}x(\sigma) = \frac{1}{2}x(T_1) + \frac{1}{2}x(T_2).$$

We now state a <u>sufficient</u> condition for tours π, σ of TP_n to be non-adjacent. (on TP_n).

<u>Sufficient condition</u> : Two tours π and σ are non-adjacent on TP_n if $|TF(\pi, \sigma)| \geq 4$ and $TF(\pi, \sigma)$ contains a pair of complementary tours.

<u>Proof</u> : If $TF(\pi, \sigma)$ contains a pair of complementary tours then π and σ are obviously non-adjacent and $|TF(\pi, \sigma)| \geq 4$.

However, the above condition is not necessary i.e. Even if $TF(\pi, \sigma)$ does not contain a pair of complementary tours, π and σ may still be non-adjacent. This is shown by the following counter example.

<u>Example 3</u> :

Let $\pi = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$, $\sigma = (1\ 5\ 9\ 4\ 8\ 3\ 7\ 2\ 6)$

Using two rowed notation, the elements of face $TF(\pi, \sigma)$ are written as below :

base row :	1 2 3 4 5 6 7 8 9
π :	2 3 4 5 6 7 8 9 1
σ :	5 6 7 8 9 1 2 3 4

$$\sigma\pi^{-1} = (2\ 5\ 8)\ (3\ 6\ 9)\ (1\ 4\ 7)$$

Elements of $TF(\pi, \sigma)$ (other than π and σ) are :

T_1 : 5 3 4 8 6 7 2 9 1 = (1 5 6 7 2 3 4 8 9)

T_2 : 2 6 4 5 9 7 8 3 1 = (1 2 6 7 8 3 4 5 9)

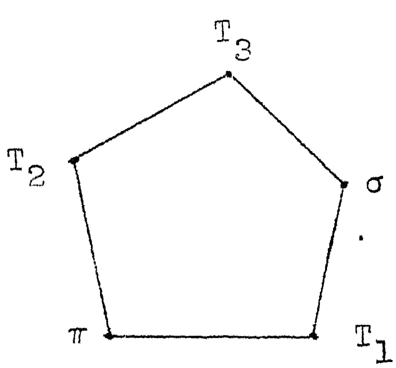
T_3 : 2 3 7 5 6 1 8 9 4 = (1 2 3 7 8 9 4 5 6)

None of the pairs (T_1, T_2), (T_1, T_3) and (T_2, T_3) is complementary w.r.t. π, σ . However, it can be observed that

$$\frac{2}{3}\, x(\pi) + \frac{1}{3}\, x(\sigma) = \frac{1}{3}\, x(T_1) + \frac{1}{3}\, x(T_2) + \frac{1}{3}\, x(T_3)$$

which establishes our claim.

Geometrically the face $TF(\pi, \sigma)$ looks like the following Figure 3.



(Figure 3)

The following theorem gives necessary and sufficient condition for non-adjacency of two tours on TP_n in the special case when $|TF(\pi, \sigma)| = 4$.

Theorem 5 : If $|TF(\pi, \sigma)| = 4$, then π and σ are non-adjacent on TP_n iff $TF(\pi, \sigma)$ contains a pair of complementary tours.

Proof : If $|TF(\pi, \sigma)| = 4$ and $TF(\pi, \sigma)$ contains a pair of complementary tours, then π and σ are obviously non-adjacent on TP_n.

Conversely let $TF(\pi, \sigma) = \{\pi, \sigma, T_1, T_2\}$ and π, σ be non-adjacent on TP_n. If T_1 and T_2 are not complementary tours, they can be obtained through two cycles S_1 and S_2 such that

$$S_1 = \bigcup_{k \in Q_1} S_k \quad , \qquad S_2 = \bigcup_{k \in Q_2} S_k$$

where Q_1 and Q_2 are subsets of $P = \{1, 2, \ldots, p\}$, assuming that $\sigma\pi^{-1}$ consists of p disjoint cycles of length greater than one, such that one of the following holds

(i) $Q_1 \cup Q_2$ is a proper subset of $\{1, 2, \ldots, p\}$

(ii) $Q_1 \cap Q_2 \neq \varphi$

We take each of the above cases separately

Case (i) Let $P^* = P - (Q_1 \cup Q_2)$, let $S^* = \bigcup_{p \in P^*} S_p$

Define $J^* = \{\pi^{-1}(i) | i \in S^*\}$

Then T_1 and T_2 are such that

$$T_1(j) = \pi(j) \quad , \quad j \in J^*$$

$$T_2(j) = \pi(j) \quad , \quad j \in J^*$$

$\Longrightarrow$ $\alpha x(\pi) + (1-\alpha)x(\sigma) \neq \beta x(T_1) + (1-\beta)x(T_2)$ for $0 \leq \alpha \leq 1$,
$0 \leq \beta \leq 1$.

$\Longrightarrow$ π, σ are adjacent which is a contradiction.

<u>Case</u> (ii) Let $S^* = \bigcup_{p \in Q_1 \cap Q_2} S_p$

Define $J^* = \left\{ \pi^{-1}(i) \mid i \in S^* \right\}$

Then T_1 and T_2 are such that

$$T_1(j) = \sigma(j) \quad , \quad j \in J^*$$

$$T_2(j) = \sigma(j) \quad , \quad j \in J^*$$

$\Longrightarrow$ $\alpha x(\pi) + (1-\alpha)x(\sigma) \neq \beta x(T_1) + (1-\beta)x(T_2)$ for $0 \leq \alpha \leq 1$,
$0 \leq \beta \leq 1$.

$\Longrightarrow$ π, σ are adjacent which is a contraction.

Thus T_1 and T_2 are complementary tours. Hence the proof.

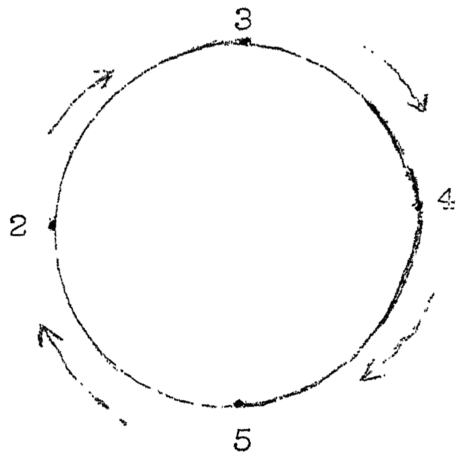
Rao [126] has also studied the properties of traveling salesman tours by using graph-theory approach and has achieved similar results.

6.3 Some Adjacency Rules and Their Applications

6.3.1. Formulation of Adjacency Rules

We formulate below some rules to generate adjacent tour to a given tour. However, many adjacency and non-adjacency rules can be formed over TP_n but it is difficult to say in any way that they are exhaustive. We are giving a brief account of a few of them only which are elementary in nature and have some applications in the development of heuristics for solving traveling salesman problem. Most of these adjacency rules are generated by permuting a subset of the elements of a tour among themselves by a suitable rule.

<u>k-cyclic change</u> : Let $\pi = (1, i_2, i_3 \ldots i_r, \overline{i_{r+1} \ldots i_{r+k}}, i_{r+k+1} \ldots i_n)$ be a tour. If k consecutive elements $i_{r+1}, i_{r+2} \ldots i_{r+k}$ as marked are chosen and replaced by the sequence $i_{r+s}, \ldots i_{r+k}, i_{r+1}, \ldots i_{r+s-1}$ where $1 < s \leq k$ then the resulting tour is called a <u>k-cyclic change of π</u>. e.g. if $\pi = (1\ \overline{2\ 3\ 4\ 5}\ 6\ 7\ 8)$ then 2 3 4 5 when replaced by 3 4 5 2, 4 5 2 3 and 5 2 3 4 gives $\sigma_1 = (1\ 3\ 4\ 5\ 2\ 6\ 7\ 8)$, $\sigma_2 = (1\ 4\ 5\ 2\ 3\ 6\ 7\ 8)$ and $\sigma_3 = (1\ 5\ 2\ 3\ 4\ 6\ 7\ 8)$ respectively. In this case each $\sigma_i (i = 1,2,3)$ is referred to as 4-cyclic change of π. Geometrically such changes can be viewed over a cycle (Figure 4) in a clockwise direction.



(Figure 4)

If the number of elements (i.e.k) is immaterial then the changes are simply referred to as cyclic changes.

<u>Rule 1</u> : Any cyclic change of a given tour π is a tour adjacent to π on TP_n.

<u>Proof</u> : Without loss , of generality, let

$$\pi = (1,2\ldots.,i,\overline{i+1\ldots i+s\ldots i+k},i_{k+1}\ldots,i_n)$$

Let σ be a k-cyclic change of π i.e.

$$\sigma = (1,2,\ldots i,\overline{i+s\ldots i+s-1},i+k+1,\ldots,n)$$

Define $K = \left\{k \mid \pi(k) \neq \sigma(k)\right\}$,

Obviously $K = \left\{i,i+s-1,i+k\right\}$

If $T = \sigma\pi^{-1}$ then we observe that

$$\begin{aligned} T(\pi(i)) &= \sigma(i) \\ &= \pi(i) \end{aligned} \qquad i \notin K$$

$$T(\pi(i)) = \sigma(i) \neq \pi(i) \qquad i \in K$$

Since a permutation over three symbols such that all the three symbols are mapped separately, is cyclic, therefore T is cyclic which implies σ is adjacent to π.

We observe that all such k-cyclic changes belong to Z_3. A special case of these cyclic changes when any two consecutive elements of π are changed is used by us to prove that TP_n is hamiltonian. (Section 6.3.2).

<u>Rule 2</u> : If a tour σ is obtained from tour π by interchanging any two non-consecutive elements then σ is adjacent to π on TP_n.

<u>Proof</u> : Without loss of generality, let

$$\pi = (1,2,\ldots,j-1,j,j+1,\ldots,k-1,k,k+1,\ldots,n)$$

Let any two non-consecutive elements, say, j and k be interchanged (1 and n are also consecutive elements). so as to get

$$\sigma = (1,2,\ldots,j-1,k,j+1,\ldots,k-1,j,k+1,\ldots,n)$$

Let $K = \{i \mid \pi(i) \neq \sigma(i)\}$

Obviously $K = \{j-1, j, k-1, k\}$

If $T = \sigma\pi^{-1}$ then we observe that

$$T(\pi(i)) = \sigma(i) = \pi(i) \qquad \text{for } i \notin K$$

and $T(\pi(i)) = \sigma(i)$

$\neq \pi(i)$ for $i \in K$

Now $T(\pi(j-1)) = \sigma(j-1)$

$\Rightarrow T(j) = k$

$T(\pi(k-1)) = \sigma(k-1)$

$\Rightarrow T(k) = j$

$T(\pi(j)) = \sigma(j)$

$\Rightarrow T(j+1) = k+1$

$T(\pi(k)) = \sigma(k)$

$\Rightarrow T(k+1) = j+1$

Thus $T = \sigma\pi^{-1}$ is product of two disjoint cycles namely (j,k) and $(j+1,k+1)$. Now by Theorem 3, π, σ are adjacent.

Tour σ as obtained above $\in Z_4$.

Rule 3 : If in a tour π any four consecutive elements are permuted among themselves then the resulting tour is adjacent to π on TP_n.

Proof : Without loss of generality, let

$$\pi = (1,2,3,\ldots \underline{k,k+1,k+2,k+3},\ldots,n)$$

Let $$\sigma = (1,2,3,\ldots \underline{s,s+1,s+2,s+3},\ldots,n)$$

where $(s,s+1,s+2,s+3)$ is some non-identity permutation of the symbols $(k,k+1,k+2,k+3)$.

Let $J = \{i \mid \pi(i) \neq \sigma(i)\}$

Then J can consist of <u>at the most</u> following five symbols

i.e. $J = \{k-1, k, k+1, k+2, k+3\}$

If $T = \sigma\pi^{-1}$ then $T(\pi(i)) = \sigma(i)$

$$= \pi(i)\ ,\ i \notin J$$

and $T(\pi(i)) = \sigma(i)$

$$\neq \pi(i)\ ,\ i \in J$$

This implies $T(\pi(i)) \neq \pi(i)$ for at most five values of i namely $i = k-1, k, k+1, k+2, k+3$.

Thus T can be a permutation of the following types only

(i) cyclic

(ii) product of two cycles such that at least one of them is of length two.

In each case σ is adjacent to π (Theorem 3).

Since what is true of tour $\pi = (1,2,\ldots,n)$ is also true of any other tour (simply the symbols are to be renamed) so for the sake of simplicity and convenience, in the following, we state results with respect to $\pi = (1,2,\ldots,n)$.

<u>Inversion</u> :

Let $\pi = (1,2,\ldots,k-1,\overline{k,k+1,k+2,\ldots,p-2,p-1,p},p+1,\ldots,n)$ be a tour.

Let $\sigma = (1,2,\ldots,k-1,\underline{p,p-1,p-2,\ldots,k+2,k+1,k},p+1,\ldots,n)$ be obtained from π such that the block $(k,k+1,k+2,\ldots,p-2,p-1,p)$ of elements appears in σ in reverse order. Then σ is

said to be obtained from π by the inversion $I(k,p)$ or simply σ is inversion of π.

The definition of inversion is due to Cröes [39].

Rule 4 : Inversion of a toue π is a tour adjacent to π on TP_n.

Proof : If $T = \sigma\pi^{-1}$ then permutation T looks like

$$T = \begin{pmatrix} k & k+1 & k+2 & k+3\ldots\ldots, & p-2 & p-1 & p & p+1 \\ p & p+1 & k & k+1\ldots\ldots, & p-4 & p-3 & p-2 & p-1 \end{pmatrix}$$

Now there are two cases (i) $p-k+1$ = even, (ii) $p-k+1$ = odd

(i) $p-k+1$ = even : In this case T is cyclic and accordingly σ is adjacent to π on TP_n.

(ii) $p-k+1$ = odd : In this case T is product of two cycles, namely

$(p,p-2,p-4,\ldots\ldots,k+4,k+2,k)$ and $(p+1,p-1,p-3,\ldots\ldots,k+3,k+1)$

$AF(\pi,\sigma)$ contains four permutations two of which are π and σ. The other two (written in two rowed notation) are :

base row : 1 2 3 .... k-1, k , k+1, k+2, ..., p-3, p-2, p-1, p , p+1, ..., n

T_1 =2 3 4 .... p , k+1, k , k+3, .., p-4, p-1, p-2, p+1, p+2, ..1

T_2 =2 3 4 .... k, p+1, k+2, k+1, .., p-2, p-3, p , p-1, p+2, ..1

It can be easily seen that both of T_1 and T_2 are non-tours. Therefore $TF(\pi,\sigma) = 2$ which implies π and σ are adjacent on TP_n.

<u>Rule 5</u> : Let $\pi = (1,2,3,\ldots,\underline{k,k+1,\ldots,k+s},\ldots r\ldots\underline{p,p+1,\ldots,p+t},\ldots,n)$
Let $(k,k+1,\ldots,k+s)$ and $(p,p+1,\ldots,p+t)$ be blocks of two elements of π separated by at least one element. Let σ be obtained from π by mutual interchange of these two blocks i.e.

$$\sigma = (1,2,3,\ldots,\underline{p,p+1,\ldots,p+t},\ldots,r,\ldots,\underline{k,k+1,\ldots,k+s},p+t+1,\ldots,n)$$

Then σ is adjacent to π on TP_n.

<u>Proof</u> : For the sake of simplicity, we first assume that $1 < k$ and $p+t<n$. However, at the end, we will observe that such restrictions are not necessary. Let

$$K = \{i \mid \pi(i) \neq \sigma(i)\}$$

Clearly $K = \{k-1, k+s, p-1, p+t\}$

If $T = \sigma\pi^{-1}$, then

$$\begin{aligned} T(\pi(i)) &= \sigma(i) \\ &= \pi(i), \quad i \notin K \end{aligned}$$

and

$$\begin{aligned} T(\pi(i)) &= \sigma(i) \\ &\neq \pi(i), \quad i \in K \end{aligned}$$

Now $T(\pi(k-1)) = \sigma(k-1)$

$$\Rightarrow \quad T(k) = p$$

$$T(\pi(p-1)) = \sigma(p-1)$$

$$\Rightarrow \quad T(p) = k$$

Similarly

$$T(k+s+1) = p+t+1$$

and $T(p+t+1) = k+s+1$

Thus T is product of two cycles, namely (k,p) and $(k+s+1, p+t+1)$. Therefore, by Theorem 3, π and σ are adjacent.

Above we have assumed that $1 < k$ and $p+t < n$. If $k = 1$ and $p+t < n$ or $k < 1$ and $p+t = n$ then there is no difficulty involved except that $p+t+1 = n+1$ should be taken as 1. However, the case $k = 1$ and $p+t = n$ needs some consideration. We observe that in this case $T(i) \neq i$ for $i = 1, p, s+2$ only and accordingly T is cyclic which implies σ is adjacent to π on TP_n.

Tour σ as obtained above $\in Z_4$ or Z_3.

Rule 6 : Let

$$\pi = (\overline{1,2,3,4,\ldots,k,\ldots,r}, r+1, r+2, \ldots, r+s_1, \ldots, r+s_2, \ldots, r+s_3, \ldots, r+s_r, \ldots, n)$$

Let the block elements $1,2,3,4,\ldots,k,\ldots,r$ be inserted in the remaining block of elements

$$(r+1, r+2, \ldots, r+s_1, \ldots, r+s_2, \ldots, r+s_3, \ldots, r+s_r, \ldots, n)$$

such that element k is inserted between $r+s_k$ and $r+s_k+1$ to give a tour σ. Then σ is adjacent to π on TP_n.

Proof :

$$\sigma = (r+1, r+2, r+3, \ldots, r+s_1, 1, r+s_1+1, \ldots, r+s_2, 2, r+s_2+1, \ldots, r+s_3, 3, r+s_3+1, \ldots, r+s_k, k, r+s_k+1, \ldots, r+s_r, r, r+s_r+1, \ldots, n)$$

Then observe that

$$\sigma\pi^{-1} = \begin{pmatrix} 2 & r+s_1+1 & 3 & r+s_2+1 & 4 & \ldots r+1 & r+s_r+1 & 1 \\ r+s_1+1 & 1 & r+s_2+1 & 2 & r+s_3+1 \ldots & r+s_r+1 & r & r+1 \end{pmatrix}$$

Clearly $\sigma\pi^{-1}$ is cyclic and the cycle is

$$(r+1, r+s_r+1, r, r+s_{r-1}+1, \ldots, 4, r+s_3+1, 3, r+s_2+1, 2, r+s_1+1, 1)$$

which is of length 2r+1.

Corollary 2 : If more than one element of the block $(1,2,3,\ldots,k,\ldots,r)$ are inserted between any two elements of the remaining block such that the order of occurence of the elements $(1,2,3,\ldots,k,\ldots,r)$ in σ is not disturbed i.e. if $j,k \in (1,2,\ldots,j,\ldots,k,\ldots,r)$ and j proceeds k then j occurs first in σ than k, then again $\sigma\pi^{-1}$ is cyclic. However, the cycle in this case is not of length 2r+1. The proof follows obviously by writing $\sigma\pi^{-1}$ as in Rule 6.

Rule 7 : Let $\pi = (1,2,3,\ldots,\overline{k,k+1,k+2,\ldots,k+p},\ldots,n)$

(i) If p = 2s(even) then construct

$$\sigma = (1,2,3,\ldots,k-1,\overline{k+1,k+3,k+5,\ldots,k+p-1,k+p,k,k+2,k+4,k+p-2},\ldots,n)$$

(ii) If p = 2s+1 (odd) then construct

$$= (1,2,3,\ldots,k-1,\overline{k+1,k+3,k+5,\ldots,k+p-2,k+p,k,k+2,k+4,\ldots,k+p-1},\ldots,n)$$

In both cases σ is adjacent to π on TP_n.

Proof : The details of the proof are omitted here as it can be proved on similar lines as the Rule 5 or Rule 6. However we give its main points.

(i) If p = 2s(even) then $\sigma\pi^{-1}$ is cyclic and the cycle is of length 2s+1.

(ii) If p = 2s+1(odd) then $\sigma\pi^{-1}$ is cyclic and the cycle is of length 2s+3.

As we have seen there can be number of rules that can be formulated to generate adjacent tours to a given tour on TP_n without exhausting all of them. Similarly there can be number of rules that can be formulated to generate non-adjacent tours to a given tour. Again it is not easy to exhaust all of them by formulating such rules. Number of such rules are formed in an attempt to separate the set of non-adjacent tours to a given tour from the set of tours adjacent to it. But always there are some tours left whose adjacency or non-adjacency cannot be decided by such systematically generated rules. We here do not list all of them but give just one rule for non-adjacency which is simple in nature and at the same time exhausts a good number of tours.

Rule 1 : Let

$\alpha = (\pi(1),\underline{\pi(2),\pi(3),\ldots,\pi(k-1)},\pi(k),\underline{\pi(k+1),\ldots,\pi(n)})$ be a tour. Let

$\beta = (\pi(1),\underline{\sigma(2),\sigma(3),\ldots,\sigma(k-1)},\pi(k),\underline{\sigma(k+1),\ldots,\sigma(n)})$

be obtained from α such that $\sigma(2), \sigma(3),\ldots, \sigma(k-1)$ and $\sigma(k+1),\ldots, \sigma(n)$ are some non-identity permutations of $\pi(2),\pi(3),\ldots,\pi(k-1)$ and $\pi(k+1),\ldots,\pi(n)$ respectively. Then β is non-adjacent to α.

Proof : Construct

$T_1 = (\pi(1), \sigma(2), \sigma(3), \ldots, \sigma(k-1), \pi(k), \pi(k+1), \ldots, \pi(n))$ and

$T_2 = (\pi(1), \pi(2), \pi(3), \ldots, \pi(k-1), \pi(k), \sigma(k+1), \ldots, \sigma(n))$

Then observe that

$$\frac{1}{2} x(\alpha) + \frac{1}{2} x(\beta) = \frac{1}{2} x(T_1) + \frac{1}{2} x(T_2)$$

which completes the proof.

6.3.2. Applications of Adjacency Rules

In the following, we describe the applications of adjacency rules for the development of heuristic procedures for solving the traveling salesman problems.

Croës [39] , to our knowledge, was the first to use the idea of permutations for tour to tour improvements for the solution of traveling salesman problems. The idea of "Inversions" is due to him. Starting with a initial tour π, an adjacent tour is generated by an inversion. (though any such adjacency ideas were not known at that time). If the objective function value for this new tour ("Inverted" tour) is less than the previous one, this "inverted" tour is retained otherwise a new inversion is tried. Similarly there are some other heuristic methods [22] which proceed from tour to tour. It is observedthat in some cases, though as a matter of coincidence only, they proceed from a tour to an adjacent tour.

Reiter and Sherman [22,129] have described a series of four similar algorithms that are called ALGOIV(r). In ALGOIV(1), one starts with a random tour, say, $\pi = (i_1, i_2, i_3, \dots, i_n)$ and finds the best tour amongest the following tours

$$\begin{aligned}
\sigma_1 &= (i_2, i_1, i_3, \dots, i_n) \\
\sigma_2 &= (i_2, i_3, i_1, \dots, i_n) \\
\sigma_3 &= (i_2, i_3, i_4, i_1, \dots, i_n) \\
&\vdots \\
\sigma_{n-2} &= (i_2, i_3, i_4, \dots, i_{n-1}, i_1, i_n)
\end{aligned}$$

(i.e. find the best position to insert i_1 in the sequence $(i_2, i_3, i_4, \dots, i_n)$). It can be observed by Rule 6 or by Rule 1 of cyclic changes that all such tours are adjacent tours. However, one is testing here only (n-2) neighbouring vertices while cyclic changes generate $\sum_{r=2}^{n-1} (r-1)(n+1-r)$ neighbouring vertices, a number much larger than (n-2).

If tour $\pi = (i_1, i_2, i_3, \dots, i_n)$ then ALGOIV(2) seeks a best possible insertion of i_1, i_2 in the sequence $(i_3, i_4, i_5, \dots, i_n)$. Although the generated tours are not in accordance with Rule 6 (since in the insertion of i_1, i_2, i_2 may proceed i_1) but still they are adjacent as $\sigma\pi^{-1}(i) \neq i$ for at most five values of i.

Similarly ALGOIV(3) seeks a best possible insertion

of (i_1, i_2, i_3) in the sequence $(i_4, i_5, i_6, \ldots, i_n)$ (assuming the current tour as $(i_1, i_2, i_3, i_4, i_5, i_6, \ldots, i_n)$). But ALGOIV(3) always does not generate adjacent tours e.g. if $\pi = (\overline{1\ 2\ 3}\ 4\ 5\ 6\ 7\ 8)$ and the tour generated is $\sigma = (4\ 2\ 5\ 1\ 3\ 6\ 7\ 8)$ then σ is non-adjacent to π. Similarly ALGOIV(4) generates some adjacent and some non-adjacent vertices..

Reiter and Sherman has coupled all the four algorithms together. When ALGOIV(1) fails to seek an improvement, the control is transferred to ALGOIV(2). After the improvement, control again returns to ALGOIV(1). If ALGOIV(1) and ALGOIV(2) both fail to give an improvement, the control is transferred to ALGOIV(3). After the improvement, control again returns to ALGOIV(1). Thus when ALGOIV(3) fails to give an improvement, control is referred to ALGOIV(4). The tour is declared optimal when ALGOIV(4) fails to give an improvement. While ALGOIV(3) and ALGOIV(4) do generate some non-adjacent vertices also, ALGOIV(1) and ALGOIV(2) to which control is referred mainly generate adjacent vertices only. However, there was no such idea of adjacency of tours even at the time of Reiter and Sherman [129].

Thus it adds to our belief that by using adjacency rules or the combinations of a few of them efficient heuristic procedures can be developed for solving traveling salesman problems.

A graph is said to be Hamiltonian if it contains a circuit which includes all nodes of the graph. Let $G(TP_n)$ be the graph of TP_n defined similar to graph $G(AP_n)$ in Chapter 5.

Theorem 6 : $G(TP_n)$ is hamiltonian.

Proof : We know that the interchange of any two consecutive elements of a tour π results in an adjacent tour (Rule 1). Using this rule and following the proof given by Balinski [12] the proof can be completed.

6.4 Further Scope of Research

We here list some problems for further research. A positive answer to these problems will contribute a lot to the traveling salesman polytope and might help in developing some algorithm for the traveling salesman problem also.

(1) To find necessary and sufficient conditions for two tours to be adjacent (or non-adjacent) on TP_n.

(2) To find the number of tours adjacent (on TP_n) to given tour. However, fair evidence is there that such a number is going to be a very large number.

(3) For a given tour π and σ such that π and σ are non-adjacent on TP_n, Padberg and Rao [121] have given a method to construct a tour T such that $T\pi^{-1}$ and

$T\sigma^{-1}$ are cyclics and accordingly T is adjacent to both π and σ on AP_n and thus also on TP_n. However, a tour T can be adjacent to both π and σ without being $T\pi^{-1}$ and $T\sigma^{-1}$ as cyclics. To exhaust the set of all those tours T such that T is adjacent to both π and σ on TP_n is another associated problem. (By taking π, σ and T as assignments one can think of similar problem in case of AP_n).

(4) If π and σ are two adjacent tours on AP_n then we know $\sigma\pi^{-1} = v$ where v is cyclic i.e. $\sigma = v\pi$ can be generated from π by the knowledge of v. Thus the problem is to find out all those possible v's (cyclics) such that generated σ's are tours. Apparently some conditions on v are needed to exhaust all those v's which generate tours adjacent to π and AP_n. Rao [126] has given one such v to construct σ adjacent to π(on AP_n).

One condition on v is that v must have odd number of elements . This is clear from the fact that $v = \sigma\pi^{-1}$ is an even permutation.

(5) Let π be the optimal assignment which is not a tour. Let $B(\pi)$ be the set of all those tours which are adjacent to π on AP_n. Then the problem is to determine the set $B(\pi)$. A few members of $B(\pi)$ are determined below :

Let π be the product of k disjoint cycles(cycles of length one are also included). We know, as shown by

Balinski [12] , also see section 5.4, that π can be written as product of two permutations one of which is a tour and the other is cycle of length k.

Thus $\pi = uv$ where u is a tour and v is cycle of length k. Clearly $u^{-1}\pi = v$ is cyclic which implies u is adjacent to π on AP_n. If $r_1, r_2, \ldots, r_k$ are the lengths of k cycles of π then possibles choices of u(or v) are

$$(r_1 r_2 \ldots . r_k)(k-1)!$$

which gives minimum number of elements of $B(\pi)$. The following result helps to some extent to know about the nature of members of $B(\pi)$.

Result : If $\sigma \in B(\pi)$ then $\sigma \in Z_p(\pi)$ for some $p \geq k$.

Because, otherwise $\sigma\pi^{-1}$ is cyclic of length $p < k$ which implies π and σ have at least one cycle common with each other. This contradicts that σ is a tour. Hence $p \geq k$.

BIBLIOGRAPHY

[1] S.P.Aggarwal " Standard Error Fractional Functional Programming", Rev-de La Facultedes Sci. Fen.Fakultesi Mathematik, Istan Univ. Turkiye, 1966.

[2] S.P.Aggarwal " Parametric Linear Fractional Functional Programming", Metrica, 12, 1968.

[3] Yuichiro Anzai " On Integer Fractional Programming", Journal of Opns.Res.Soc.of Japan, 17, 1964, pp.49-66.

[4] S.R.Arora and M.C.Puri " Enumeration Technique for the Set Covering Problem with Linear Fractional Functional as its Objective Function", ZAMM, 57, 1977, pp.181-186.

[5] K.J.Arrow and A.C.Enthoven " Quasi Concave Programming", Econometrica, 29, 1961, pp.779-800.

[6] K.J.Arrow, L. Hurwicz and H.Uzawa " Studies in Linear and Non-Linear Programming", Standford University Press, 1958.

[7] K.J.Arrow, L. Hurwicz and H.Uzawa " Constraint Qualifications in Maximization Problems", NRLQ, 8, 1961, pp.175-191.

[8] Egon Balas " An Additive Algorithm for Solving Linear Programs with Zero-One Variables", Opns. Res., 13, 1965, pp.517-546.

[9] Egon Balas and M.W.Padberg " On the Set Covering Problem", Opns.Res., 20, 1972, pp.1152-1161.

[10] Egon Balas and M.W.Padberg " On the Set Covering Problem II, An Algorithm for Set Partitioning", Opns.Res., 1975, pp.74-90.

[11] M.L.Balinski and R.E.Gomory " A Primal Method for the Assignment and Transportation Problems", Man.Sci.,10,1964, pp.578-593.

[12] M.L.Balinski and A.Russakoff " On the Assignment Polytope", SIAM Review,16,1974, pp.516-525.

[13] E.M.L.Beale " On Minimizing a Convex Function subject to Linear Inequalities", Jour. Roy.Stat.Soc.(B),17, 1955, pp.173-184.

[14] E.M.L.Beale " On Quadratic Programming", NRLQ,6,1959,pp.227-243.

[15] C.R.Bector " Programming Problems with Convex Fractional Functionals", Opns.Res.,16,1968,pp.383-391.

[16] C.R.Bector " Duality in Fractional and Indefinite Programming", ZAMM,48,1968.

[17] C.R.Bector and S.K.Bhatt " Pseudo-monotonic Interval Programming",Appeared in NRLQ (in 1977 or 1978).

[18] C.R.Bector and P.L.Jolly " Pseudo-monotonic Integer Programming", Proceedings of the Seventh Manitoba Conference on Numerical Mathematics and Computing, 1977,pp.211-218.

[19] C.R.Bector and P.L.Jolly " Programming Problems with Pseudo-monotonic Objectives", accepted (Germany).

[20] C.R.Bector and P.L.Jolly " Sensitivity Analysis of Linear Complementary Problems with Applications", presented in Joint National Meeting of ORSA/TIMS held at Los Angeles, U.S.A. on Nov.13-15,1978 and abstracted as TAA 3.2 ORSA/TIMS Bulletin 1978.

[21] R.Bellman " Notes on the Theory of Dynamic Programming IV-Maximization over Discrete Sets", NRLQ,3,1956,pp.67-70.

[22] M.Bellmore and G.Nemhauser " The Traveling Salesman Problem: A Survey", Opns.Res.,16,1968, pp.538-558.

[23] M.Bellmore and H.Ratliff " Set Covering and Involutory Bases", Man.Sci.,18,1971, pp.194-206.

[24] A.Ben-Israel and A.Charnes " On Some Problems of Diophantine Programming", Cah.du Cen.d' Etudes de Rec. Oper.,4,1962,pp.215-280.

[25] S.K.Bhatt " Generalized Pseudo-Convex Programming and Optimal Control",Thesis Submitted to the Deptt. of Math.,I.I.T. Kanpur,India,1971.

[26] E.K.Blum " Minimization of Functionals with Inequality Constraints", J.SIAM Control,3,1965,pp. 299-316.

[27] J.C.G.Boot " On Sensitivity Analysis in Convex Quadratic Programming Problems", Opns.Res,11,1963, pp.771-786.

[28] Suresh Chandra " Duality in Banach Spaces; Transportation, Assignment and Large Structured Fractional Programs", Thesis Submitted to Deptt. of Maths. IIT,Kanpur,India,1969.

[29] R.Chandrasekaran " Total Unimodularity of Matrices", SIAM J. Appl.Math., 17,1969,1032-1034.

[30] A.Charnes and W.W.Cooper " The Stepping Stone Method of Explaining Linear Programming Calculations in Transportation Problems", Man.Sci.,1,1954, pp.49-69.

[31] A.Charnes and W.W.Cooper " Programming with Linear Fractional Functions", NRLQ, 9, 1962, pp.181-186.

[32] A.Charnes, W.W. Cooper and K.O.Kortanek " Duality in Semi-infinite Programs and some works of Haar and Caratheodary", Man. Sci., 9, 1963, pp.209-228.

[33] A.Charnes, W.W. Cooper and K.O.Kortanek " On Representation of Semi-infinite Programs which have no Duality Gaps", Man.Sci., 12, 1965, pp.113-121.

[34] A.Charnes, W.W. Cooper and K.O.Kortanek " On the Theory of Semi-finite Programming and generalization of Kuhn-Tucker Saddle Point Theorem for arbitrary Convex Functions", NRLQ, 16, 1969, pp.41-51.

[35] A.Charnes, W.W. Cooper and K.O.Kortanek " Semi-infinite Programming, Differentiability and Geometric Programming, Part II", Aplikace Mathematicky (Czechoslovakia), 14, 1969.

[36] Richard W.Conway, W.L.Maxwell and L.W.Miller. " Theory of Scheduling", Addison Wesley, 1967 (Three Machine Scheduling, pp.93).

[37] R.W.Cottle and G.B.Dantzig " Complementary Pivot Theory of Mathematical Programming", Mathematics of Decision Sciences, edited by G.B.Dantzig and A.F. Veinott, AMS, 1968, pp.115-136.

[38] R.W.Cottle and G.B.Dantzig " Complementary Pivot Theory of Mathematical Programming", Linear Algebra and its Applications, 1, 1968, pp.103-125.

[39] G.Croës " A Method for Solving Traveling Salesman Problems", 6, 1958, pp. 791-812.

[40] G.B.Dantzig " Upper Bounds, Secondary Constraints and Block Triangularity in Linear Programming", Econometrica, 23, 1955, pp.174-183.

[41] G.B.Dantzig " Linear Programming and Extensions", Princeton Univ. Press, N.J.,1963.

[42] G.B.Dantzig and R.W.Cottle " Positive (Semi) definite Programming", Non-Linear Programming edited by J.Abadie, North Holland, Amsterdam,1967, pp.55-73.

[43] G.B.Dantzig D.R.Fulkerson and S.Johnson " Solution of a Large Scale Travelling Salesman Problem", Opns.Res., 2,1954,pp.393-410.

[44] W.Dinkelbach " On Nonlinear Fractional Programming", Man.Sci.,13,1967, pp.492-498.

[45] W.S.Dorn " Linear Fractional Programming", IBM Research Report 1962.

[46] R.J.Duffin " Dual Programs and Minimum Cost", J.SIAM,10,1962,pp.119-123.

[47] R.J.Duffin and L.A.Kartovitz " An Infinite Linear Program with a Duality Gap", Man.Sci., 12,1965,pp.122-134.

[48] R.J.Duffin and E.L.Peterson " Duality Theory for Geometric Programming", SIAM J. Appl. Math.,14,1966,pp.1307-1349.

[49] W.Eastman " Linear Programming with Pattern Constraints", Ph.D. Thesis, Havard Univ.,1958.

[50] B.C.Eaves " The Linear Complementarity Problems", Man.Sci.,17,1971, pp.612-634.

[51] B.C.Eaves " On Quadratic Programming", Man.Sci.,17,1971,pp.698-711.

[52] J.Etcheberry " The Set Covering Problems : A New Implicit Enumeration Algorithm", Opns.Res.,25,1977, pp.760-772.

[53] M.Florian and P.Robillard " Programmation Hyperbolique en Variables Bivalents", Revue Francaise d' Informatique et de Rech.Oper., 1, 1971, pp.3-9.

[54] L.R.Ford and D.R.Fulkerson " Solving the Transportation Problem", Man.Sci., 3, 1956, pp.24-32.

[55] L.R.Ford and D.R.Fulkerson " A Primal-Dual Algorithm for the Capacitated Hitchcock Problem", NRLQ, 4, 1957, pp.47-54.

[56] R.S.Garfinkel " Minimizing Wall Paper Waste, Part 1 : A Class of Traveling Salesman Problem", Opns.Res., 25, 1977, pp.741-751.

[57] R.S.Garfinkel and G.L.Nemhauser " The Set Partitioning Problem : Set Covering with Equality Constraints", Opns.Res., 17, 1969, pp.848-856.

[58] R.S.Garfinkel and G.L.Nemhauser " Integer Programming", John Wiley, N.Y., 1972.

[59] S.I.Gass " Linear Programming" - Methods and Applications", McGraw-Hill Kogakusha Ltd, 1969.

[60] A.M.Geoffrion " Integer Programming by Implicit Enumeration and Balas' Method", SIAM Review, 9, 1967, pp.178-190.

[61] F. Glover " A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem", Opns.Res., 13, 1965, pp.879-919.

[62] F. Glover " An Algorithm for Solving the Linear Integer Programming Problem over a Finite Additive Group, with Extensions to Solving General and Certain Nonlinear Integer Programs", Opns.Res. Centre Report 66-29, The Univ. of California at Berkely, Sept. 1966.

[63] F.Glover " A Pseudo Primal-Dual Integer Programming Algorithm", Jour. of Res. of National Bureau of Standards, 71B, 1967, pp.187-195.

[64] F.Glover " A New Foundation for a Simplified Primal Integer Programming Algorithm", Opns. Res., 16, 1968, pp.727-740.

[65] N.K.Gogia " The Multiplex Method for Linear Fractional Functional Programming", Cah. du Centre d' Etudes de Rech. Oper., 9, 1967.

[66] R.E.Gomory " An Algorithm for Integral Solutions to Linear Programming", Princeton-IBM Mathematics Research Project, Tech.Rep.No.1, Nov.1958.

[67] R.E.Gomory " An Algorithm for the Mixed Integer Problem", P-1885, The RAND Corp., June 1960.

[68] R.E.Gomory " On the Relation between Integer and Non Integer Solutions to Linear Programs", Proceedings of the National Academy of Science, 53, 1965, pp.260-265.

[69] R.L.Graves " A Principal Pivoting Simplex Algorithm to Linear and Quadratic Programming", Opns. Res.15, 1967, pp.482-494.

[70] M.Grignard " Generalized Kuhn-Tucker Condition for Mathematical Programs in Banach Space", J.SIAM Control, 7, 1969.

[71] B.Grünbaum " Convex Polytopes", Wiley, N.Y., 1966.

[72] M.Grunspan and M.E.Thomas " Hyperbolic Integer Programming", NRLQ, 20, 1973, pp.341-356.

[73] S.K.Gupta and S.K.Bhatt " Generalized Pseudo-Convex Functions", Cah. du Cen. d' Etudes de Res.Oper., 11,1971.

[74] G.Hadley " Nonlinear and Dynamic Programming", Addison-Wesley, 1972.

[75] G.Hadley " Linear Algebra",Add-Wesley,1973.

[76] G.Hadley " Linear Programming",Add-Wesley, 1973.

[77] M.A.Hanson " Infinite Non-Linear Programming", J.Aust.Math.Soc.,3,1963, pp.294-300.

[78] F.Harary " Graph Theory", Add-Wesley, 1969.

[79] M.Held and R.Karp " The Travelling Salesman Problem and Minimum Spanning Trees", Opns.Res.18,1970,pp.1138-1162.

[80] M.Held and R.Karp " The Travelling Salesman Problem and Minimum Spanning Trees-Part II", Math . Prog.,1,1971, pp.6-25.

[81] H.S.Houthakker " The Capacity Method for Quadratic Programming", Econometrica.28,1960,pp.62-87.

[82] L.Hurwicz " Programming in Linear Spaces", Studies in Linear and Non-Linear Programming, Standford Univ. Press,1958.

[83] T.Ibaraki " Complementary Programming", Opns.Res.,19,1971,pp.1523-1529.

[84] T.Ibaraki " The Use of Cuts in Complementarity Programming", Opns.Res., 21,1973,pp.353-359.

[85] H.Ishii,T. Ibaraki and H.Mine " A Primal Cutting Plane Algorithm For Integer Fractional Programming Problems", Jour. of the Opns.Res.Soc. of Japan,19, 1976,pp.228-244.

[86] P.A.Jenson " Optimum Network Partitioning", Opns.Res.,19,1971,pp.916-932.

[87] R.G.Jeroslow " Cutting-planes for Complement-arity Constraints", SIAM Control and Optimization,16,1978, pp.56-62.

[88] Fritz John " Extremum Problems with Inequalities as Side Conditions", in Studies and Essays, Courant Anniversary Volume,pp.187, John Wiley(INC), N.Y.1948.

[89] P.Jolly " Connectivity on the Assignment Polytope", Presented at the 9th Annual Convention of Opns. Res.Soc.of India held at Calcutta on Dec.3-5,1976(and was awarded the prize for the best student paper).

[90] P.Jolly " A Method for Bounded Variable Linear Programs", Presented at Annual Meeting of Bharat Ganit Parishad held at Lucknow (India) on 8th April,1979.

[91] P.Jolly " A Step by Step Dual Simplex Approach for Bounded Variable Linear Programs", Presented at Annual Meeting of Bharat Ganit Parishad held at Lucknow(India) on 8th April,1979.

[92] Samual Karlin " Mathematical Methods and Theory in Games, Programming, and Economics", Add-Wesley, 1959 (pp.132-133, Chapter 5).

[93] R.M.Karp " Reducibility among Combinatorial Problems", in Complexity of Computer Computations", R.E. Miller and J.W.Thatcher, Eds., Plenum Press,N.Y.1972,pp.85-103.

[94] R.J.Kelley and W.A.Thomson " Quadratic Programming in Real Hilbert Space", J.SIAM Appl. Math.,11,1963,pp.1063-1070.

[95] K.Kirchgässnei " Einverfahren Zur Maximierung Linear Furktionen in Nichtkonvexen Bereichen", Zeits. Angewandte Mathematik, 1962,T 22-24.

[96] V.Klee and C.Witzgall " Facets and Vertices of Transportation Polytopes", Math. of the Decision Sciences, Vol.II,edited by G.B.Dantzig and A.F.Veinott, AMS 1968, pp.257-282.

[97] K.O.Kortanek and J.P.Evans " Pseudo-Concave Programming and Lagrange Regularity", Opns.Res.,15,1967,pp.882-891.

[98] H.W.Kuhn " The Hungarian Method for the Assignment Problem", NRLQ,2, 1955,pp.83-97.

[99] H.W.Kuhn and A.W.Tucker " Non-Linear Programming", Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability", 1951,pp.481-492.

[100] A.Land and A.Doig " An Automatic Method of Solving Discrete Programming Problem", Econometrica 28, 1960,pp.497-520.

[101] E.Lawler " Covering Problems : Duality Relations and a New Method of Solution",SIAM J. Appl.Maths. 14,1966,pp.1115-1132.

[102] Harvey Lee " An Experimental Study on Solving Convex Quadratic Programming Problems", Unpublished M.S. Project, School of Industrial Engg.,Purdue Univ.,Indiana , August 1975.

[103] C.E.Lemke " The Dual Method of Solving the Linear Programming Problem", NRLQ,1,1954,pp.48-54.

[104] C.E.Lemke " Bimatrix Equilibrium Points and Mathematical Programming", Man.Sci.,4,1965,pp.681-689.

[105] C.E.Lemke " Recent Results on Complementarity Problems", in Nonlinear Programming Edited by J.B. Rosen, O.L.Mangasarian, K.Ritter, Proceedings of a Symposium Conducted by the Mathematic Research Centre, The Univ. of Winconsin, Madison, 1970, pp. 349-384.

[106] C.E.Lemke and J.T.Howson " Equilibrium Points of Bimatrix Games", SIAM J. Appl.Math., 12, 1964, pp.413-423.

[107] C.E.Lemke, H.Salkin and K.Spielberg " Set Covering by Single Branch Enumeration with Linear Programming Sub Problems", Opns.Res., 19, 1971, pp.998-1022.

[108] J.K.Lenstra " Some Simple Applications of the Traveling Salesman Problem", Opns.Res. Quart., 26, 1975, pp.717-733.

[109] J.Little, K.Murty, D.Sweeney and C.Karel " An Algorithm for the Travelling Salesman Problem", Opns.Res., 11, 1963, pp.972-989.

[110] G.Maier " A Matrix Structural Theory of Piece Wise Linear Elasto-Plasticity with Interactive Yield Planes", Meccanica, 5, 1970, pp.54-66.

[111] O.L.Mangasarian " Pseudo-Convex Functions", J.SIAM Control, 3, 1965, pp.281-290.

[112] O.L.Mangasarian " Nonlinear Programming", McGraw Hill, 1969.

[113] O.L.Mangasarian and S.Fromovitz " The Fritz John Necessary Optimility Conditions in the presence of Equality and Inequality Constraints", Jour. Math.Analys.Appl., 7, 1967, pp.37.

[114] Bela Martos " Hyperbolic Programming", NRLQ, 11, 1964, pp.135-155 (Formerly appeared in Mathematical Institute, Hungarian Acad. of Sci., 5, 1960, pp.383-406.

[115] Bela Martos " The Direct Power of Adjacent Vertex Programming Methods", Man.Sci.,12,1965,pp. 241-252.

[116] Bela Martos " Nonlinear Programming : Theory and Methods", North Holland, Amsterdam,1975.

[117] James Munkres " Algorithm for the Assignment and Transportation Problems", J.SIAM,5,1957,pp.32-38.

[118] K.G.Murty " Adjacency on Convex Polyhedra", SIAM Review, 13,1971,pp.377-386.

[119] K.G.Murty " Linear and Combinatorial Programming", John Wiley,1976.

[120] L.W.Neustadt " Optimization, a Moment Problem and Non-Linear Programming", J.SIAM Control,2,1964.

[121] M.W.Padberg and M.R.Rao " The Travelling Salesman Problem and a class of Polyhedra of Diameter Two", Math.Prog.,7, 1974,pp.32-45.

[122] U.Passy and D.J.Wilde " Generalized Polynomial Optimization", SIAM J. Appl. Math.,15,1967,pp.1344-1356.

[123] B.N.Pchenichney " Linear Optimal Control", J.SIAM Control,4,1966.

[124] D.T.Philips,A. Ravindram, and James Solberg " Operations Research,Principles and Practice", John Wiley, 1976.

[125] J.Polito " A Comparision of Four Versions of a Modified Cottle-Dantzig Algorithm with the Lemke Method on Convex Quadratic Programs", Unpublished M.S.Thesis,School of Industrial Engg.,Purdue Univ., Indiana, Dec.1974.

[126] M.R.Rao " Adjacency of the Travelling Salesman Tours and 0-1 Vertices", SIAM J. Appl.Math.,30,1976, pp.191-198.

[127] A.Ravindran " Computational Aspects of Lemke's Compl&mentary Algorithm Applied to Linear Programs", Opsearch,7,1970, pp.241-262.

[128] A.Ravindran " A Comparision of Primal Simplex and Compl&mentary Pivot Methods for Linear Programming", NRLQ, 20,1973,pp. 95-100.

[129] S.Reiter and G.Sherman " Discrete Optimizing",SIAM,13, 1965,pp.864-889.

[130] C.Revelle,D. Marks and J. Liebman " An Analysis of Private and Public Sector Location Models", Man.Sci.,16,1970,pp.692-707.

[131] K.Ritter " Duality for Non-Linear Programming in Banach Space", J.SIAM Appl.Math.,15,1967, pp.294-302.

[132] D.J.Rosenkrantz R.E.Stearns and P.M.Lewis II " Approximation Algorithms for the Travelling Salesperson Problem", Report No.76 CRD266, Nov.1976,Automatic and Control Laboratory, General Electric Company, Corporate Research and Development, Schenectady,N.Y.

[133] J.Rosenmüller " On a Generalization of Lemke-Howson Algorithm to Non-Cooperative n-person Games", SIAM J. Appl.Math.,21,1971, pp.73-79.

[134] Ramesh Saigal " On the Class of Compl&mentary Cones and Lemke's Algorithm," SIAM J.Appl.Math.,23,1972, pp.46-60.

[135] H.Salkin " Integer Programming", Add-Wesley, 1975.

[136] H.Salkin and R.Koncal " A Pseudo-Dual All Integer Algorithm for the Set Covering Problem", Deptt. of Opns.Res.Tech.Memorandum No. 204,Case Western Reserve Univ., Nov.1970.

[137] H.Salkin and R.Koncal " A Dual All Integer Algorithm (In Revised Simplex Form) for the Set Covering Problem", Deptt. of Opns..Res.Tech. Memorandum No. 250,Case Western Reserve Univ.,August1971.

[138] H.Scarf " The Core of an n-person Games", Econometrica,35,1967,pp.50-69.

[139] H.Scarf " The Approximation of Fixed Points of a Continuous Mapping", SIAM J. Appl.Math., 15,1967, pp.1328-1342.

[140] H.Scarf and T.Hansen " On the Applications of a Recent Combinatorial Algorithm", Cowles Foundation Discussion Paper No.272,Yale Univ.April1969.

[141] Kanti Swarup " Some Aspects of Linear Fractional Functional Programming", Aust.Jour. of Statistics,7,1965, pp.90-104.

[142] KantiSwarup " Linear Fractional Functional Programming", Opns.Res.,13,1965, pp.1029-1036.

[143] Kanti Swarup " Transportation Technique in Linear Fractional Functional Programming",J.R.N.S.S.(U.K.), 21,1966,pp.256-260.

[144] Kanti Swarup " Fractional Programming with Nonlinear Constraints",ZAMM (Germany),46,1966.

[145] Kanti Swarup " Indefinite Quadratic Programming", Cah. du Cen. d' Etudes Rech. Oper.,8,1966.

[146] Kanti Swarip " Programming with Indefinite Quadratic Function with Linear Constraints", Cah.du Cen.d' Etudes Rech.Oper.,8,1966.

[147] H.A.Taha " An Algorithm for Zero-One Fractional Programming", AIIE, Transactions,7,1975,pp.29-34.

[148] H.A.Taha "Operations Research, An Introduction", Macmillian,1976.

[149] C.Toregas, R. Swain, C.Revelle and L.Bergman "The Location of Emergency Service Facilities", Opns.Res., 19, 1971, pp.1363-1373.

[150] C.Van de-Panne and A.Whinston "The Simplex and the Dual Method for Quadratic Programming", Opns.Res.Quart., 15, 1964, pp.355-388.

[151] C.Van de-Panne and A.Whinston "Simplicial Methods for Quadratic Programming", NRLQ, 11, 1964, pp. 273-302.

[152] C.Van de-Panne and A.Whinston "The Symmetric Formulation of the Simplex Method for Quadratic Programming", Econometrica, 37, 1969, pp.507-527.

[153] P.Varaiya "Nonlinear Programming in Banach Space". J.SIAM Appl.Math., 15, 1967, pp.284-293.

[154] R.Wilson " Computing Equilibria of n-person Games", SIAM J Appl.Math., 21, 1971, pp.80-87.

[155] P.Wolfe "The Simplex Method for Quadratic Programming", Econometrica, 27, 1959, pp.382-398.

[156] R.Young "A Primal (All Integer) Integer Programming Algorithm", Journal of Research of the National Bureau of Standards, 69B, 1965, pp.213-249.

[157] R.Young "A Simplified Primal (All Integer) Integer Programming Algorithm", Opns.Res.16, 1968, pp.750-782.

[158] W.I.Zangwill "The Piecewise Concave Functions", Man.Sci., 13, 1967, pp.900-912.